

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR  
FACULTAD DE INGENIERÍA  
ESCUELA DE SISTEMAS**

**DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN SISTEMAS Y COMPUTACIÓN**

**ANÁLISIS, DESARROLLO, IMPLEMENTACIÓN E  
IMPLANTACIÓN DE UN SISTEMA PARA LA AUTOMATIZACIÓN  
DE PROCESOS EN UN CENTRO MÉDICO. CASO DE ESTUDIO:  
E.M. CENTRO DE ESPECIALIDADES MÉDICAS, UTILIZANDO  
LA METODOLOGÍA EXTREME PROGRAMMING (XP).**

**CARLOS ENCALADA**

**DIRECTOR: ING. JORGE ALARCÓN**

**Quito, 2018**

## **AGRADECIMIENTOS**

A Dios, por acompañarme siempre en todos mis planes, sin importar lo audaces que fueran; y por enseñarme que cuando la vida deja de creer en uno, es cuando uno debe creer más en sí mismo.

## Tabla de Contenidos

Índice de Tablas .....	IV
Índice de Figuras .....	IV
Capítulo 1 – Aspectos Introdutorios .....	1
1.1 Datos de la Organización .....	1
1.2 Justificación .....	2
1.3 Planteamiento del Problema .....	2
1.4 Alcance .....	3
1.5 Objetivos .....	3
1.5.1 Objetivo General .....	3
1.5.2 Objetivos Específicos .....	3
Capítulo 2 – Fundamento Teórico .....	4
2.1 Metodología de Desarrollo .....	4
2.1.1 Análisis .....	4
2.1.2 Extreme Programming .....	4
2.2 Herramientas .....	6
2.2.1 Base de Datos .....	6
2.2.2 Lenguajes de Programación .....	7
2.2.3 Marcos de Trabajo .....	8
Capítulo 3 – Caso de Estudio .....	11
3.1 Historias de Usuario .....	11
3.2 Planificación .....	16
3.2.1 Información complementaria .....	17
3.2.2 Casos de Uso .....	17
3.2.3 Formalización de Requerimientos .....	23
3.3 Diseño .....	24
3.3.1 Modelo Conceptual .....	25
3.3.2 Estructura de Base de Datos .....	26
3.3.3 Arquitectura .....	27
3.4 Codificación .....	27
3.4.1 Artisan Console .....	28
3.4.2 Migraciones .....	28
3.4.3 Datos de Prueba .....	28

3.4.5 Esquema de Interfaces.....	30
3.5 Pruebas.....	31
Capítulo 4 – Otras Iteraciones.....	32
4.1 Versiones .....	32
4.2 Prototipos .....	33
4.3 Pruebas de Aceptación.....	35
4.4 Rutas .....	39
4.5 Controladores.....	41
4.6 Vistas .....	41
Capítulo 5 – Implantación y Aceptación .....	44
5.1 Pruebas Unitarias .....	44
5.1.1 Paciente .....	44
5.1.2 Asistente .....	46
5.1.3 Doctor.....	49
5.1.4 Administrador.....	51
5.2 Aceptación .....	53
Capítulo 6 –Conclusiones y Recomendaciones .....	54
6.1 Conclusiones .....	54
6.2 Recomendaciones .....	54
Glosario.....	55
Referencias.....	58
Anexos .....	59
Actas de Reunión .....	59
Documento de Aceptación.....	66
Manual Técnico .....	1
Introducción .....	1
Roles.....	1
Software de Terceros.....	3
Diccionario de Datos.....	3
Instalación y configuración .....	5
Manual de usuario.....	1
Pantalla inicial .....	1
Registro de usuario.....	2
Inicio de sesión.....	3

Opciones para los roles .....	4
Respaldar datos .....	4
Configurar parámetros.....	4
Gestión de Roles.....	5
Gestión de Horarios.....	6
Gestión de Usuarios .....	7
Gestión de Citas .....	9
Opciones del sistema.....	10

## Índice de Tablas

Tabla 1 Datos de la Organización.....	1
Tabla 2 Historia de Usuario SD01.....	13
Tabla 3 Historia de Usuario SO01.....	14
Tabla 4 Historia de Usuario SG01.....	15
Tabla 5 Historia de Usuario SA01.....	15
Tabla 6 Requerimientos funcionales .....	23
Tabla 7 Requerimientos no funcionales .....	24
Tabla 8 Requerimientos mínimos.....	24
Tabla 9 Detalle de migraciones. ....	28
Tabla 10 Generación de datos de prueba.....	29
Tabla 11 Pruebas unitarias iniciales. ....	31
Tabla 12 Rutas no protegidas. ....	39
Tabla 13 Recursos, Controladores y Filtros. ....	39
Tabla 14 Rutas específicas. ....	40
Tabla 15 Detalle de vistas.....	43
Tabla 16 Pruebas unitarias para pacientes.....	44
Tabla 17 Pruebas unitarias para asistentes. ....	46
Tabla 18 Pruebas unitarias para doctores. ....	49
Tabla 19 Pruebas unitarias para administradores. ....	51

## Índice de Figuras

Figura 1 Planificación Inicial .....	16
Figura 2 Diagrama general de casos de uso .....	18
Figura 3 Gestión de usuarios .....	19
Figura 4 Gestión de pacientes.....	19
Figura 5 Gestión de doctores.....	20
Figura 6 Gestión de contactos .....	20
Figura 7 Gestión de registros.....	21

Figura 8 Gestión de citas médicas. ....	21
Figura 9 Gestión de horarios de doctores. ....	22
Figura 10 Gestión de parámetros. ....	22
Figura 11 Diseño conceptual. ....	25
Figura 12 Diseño de base de datos. ....	26
Figura 13 Arquitectura del sistema. ....	27
Figura 14 Esquema de interfaz gráfica. ....	30
Figura 15 Esquema de iteraciones (pt. 1) ....	32
Figura 16 Esquema de iteraciones (pt. 2) ....	33
Figura 17 Prototipo de indexado. ....	34
Figura 18 Prototipo de perfil. ....	34
Figura 19 Prototipo de creación y edición. ....	35

## Capítulo 1 – Aspectos Introductorios

El desarrollo de proyectos de software se lleva a cabo en contextos muy variados con el objetivo de atender necesidades muy numerosas, pero al mismo tiempo particulares, en el campo de la medicina, especialmente, existen procesos muy simples, pero al mismo tiempo muy importantes; procesos como la administración de citas o la administración de historiales que son comunes a través de organizaciones de salud tanto públicas como privadas. En este capítulo se tratan los asuntos referentes al contexto inicial de la disertación, dando especial importancia a los aspectos que determinan la necesidad de llevar a cabo el presente trabajo.

La presente disertación contiene los siguientes capítulos:

- Aspectos Introductorios, donde se trata acerca de la motivación y los objetivos del proyecto.
- Fundamento teórico, donde se describen los aspectos relacionados a la base teórica subyacente sobre la cual se sustenta el proyecto.
- Caso de Estudio, donde se analiza y posteriormente formaliza de forma ordenada los requerimientos del proyecto y se plantea una estrategia para abordarlos.
- Otras Iteraciones, trata acerca de las etapas de desarrollo adicionales que fueron realizadas para cumplir los objetivos.
- Implantación y Aceptación, en este capítulo se trata acerca de la realización de pruebas sobre el producto de manera que se puedan validar los requerimientos del proyecto.
- Conclusiones y Recomendaciones, en esta sección se realiza un contraste de los objetivos y requerimientos del proyecto con respecto al resultado final obtenido.

### 1.1 Datos de la Organización

<b>Nombre</b>	E.M. Centro de Especialidades Médicas
<b>Actividad</b>	Salud, atención a pacientes.
<b>Ubicación</b>	Humberto Albornoz Oe8-181 y Atacames, Quito - Ecuador.
<b>Características</b>	Seguimiento de historiales y atención a pacientes.

Tabla 1 Datos de la Organización  
Encalada, C. (junio 2018).

## **1.2 Justificación**

La información como recurso importante y sensible en una organización debe ser administrada y presentada en forma apropiada. A menudo las PYMES (pequeñas y medianas empresas) no cuentan con un sistema que permita manejar su información de forma apropiada.

Esto es especialmente cierto para organizaciones en el sector de la salud que por su tamaño entran en la categoría de PYMES, las herramientas informáticas utilizadas en este ámbito particular a menudo tienen un alto grado de especialización, sin embargo, no manejan estándares que les permitan trabajar de manera conjunta.

Resulta necesario aclarar cuáles son los procesos que se manejan dentro de una organización que funciona bajo las características expuestas anteriormente, y de forma particular, el caso de estudio del Centro de Especialidades Médicas, para así poder desarrollar una herramienta unificada que facilite el manejo de la información de manera segura y efectiva a nivel interno.

## **1.3 Planteamiento del Problema**

El Centro de Especialidades Médicas, debe manejar el historial médico de aproximadamente 400 pacientes, para el efecto cuenta con un equipo informático que mantiene una colección de archivos poco estructurada y gestionada por un software desactualizado cuyo versionamiento ha sido discontinuado por el fabricante.

Adicional a esto, dado que una parte importante de los pacientes son atendidos por el área de pediatría, el personal requiere realizar evaluaciones expedientes que involucran cálculos específicos; estos se vienen realizando con otras herramientas de software que tienen un funcionamiento particular y que requieren de la selección manual de los datos de pacientes.

El Centro de Especialidades Médicas también entrega informes de diagnóstico e historial a petición de los pacientes, proceso que se realiza de forma manual dado que el componer un informe de cualquier naturaleza requiere de explorar los archivos previamente almacenados para luego juntarlos en un documento.



## **1.4 Alcance**

El presente trabajo de disertación se considerará terminado una vez que se haya implantado el sistema con los requerimientos levantados en las correspondientes etapas para lo cual el Centro de Especialidades Médicas extenderá una carta de entrega-recepción a plena satisfacción; a ello se sumarán todas las sugerencias emitidas por el Director y los Correctores de la Disertación.

## **1.5 Objetivos**

### **1.5.1 Objetivo General**

- Construir e Implantar un Sistema para la Automatización de Procesos en el Centro de Especialidades Médicas.

### **1.5.2 Objetivos Específicos**

- Analizar los requerimientos para el desarrollo e implementación del sistema.
- Diseñar una arquitectura de sistema que se ajuste a los requerimientos del proyecto.
- Desarrollar el sistema en base a módulos.
- Implantar el sistema.

## **Capítulo 2 – Fundamento Teórico**

Este capítulo trata acerca de la metodología de desarrollo utilizada y las herramientas que acompañaron el proceso. En el desarrollo de software, es muy importante basar el trabajo a realizar sobre una metodología, esto se debe a que las metodologías de desarrollo presentan un marco de trabajo cuyo objetivo es definir los pasos a seguir, ayudar a controlar el tiempo de desarrollo y apoyar en la generación de documentación sobre la cual sustentar el trabajo realizado.

### **2.1 Metodología de Desarrollo**

La metodología de desarrollo se refiere al marco de trabajo sobre el cual se estructura, planifica y controla el proceso de desarrollo de un componente de software o sistema informático.

#### **2.1.1 Análisis**

Para este caso de estudio en particular se tiene un contexto en el que existe la participación activa del cliente, quien no tiene una idea firme sobre lo que el sistema debería hacer y los procesos que debería manejar, por lo cual se asume que los requerimientos podrían tener variaciones. El desarrollo deberá ser sometido a evaluación periódica a cargo del director del proyecto considerando debidamente el cronograma y las metas que deben ser cumplidas por exigencia de la universidad.

Se escogió Extreme Programming (XP) porque tiene como objetivo generar eficiencia en el proceso de codificación y proporcionar a los clientes el máximo valor al tiempo que facilita la comunicación de éste con el equipo de desarrollo. Las ventajas de XP incluyen robustez, resistencia, ahorro de costos, gestión de riesgos y una mejor satisfacción general; asimismo XP facilita la realización de evaluaciones sobre proyectos en curso sin que esto interfiera con el desarrollo maximizando así la tolerancia a los cambios que pudiesen surgir.

#### **2.1.2 Extreme Programming**

Extreme Programming (XP) es una metodología ágil popular, desarrollada a base de principios del sentido común y prácticas habituales llevadas a niveles extremos. Similar a sus metodologías hermanas, XP promueve el versionamiento a partir de porciones funcionales

de código con la finalidad de incrementar la productividad y mantener el flujo apegado a los requerimientos con especial énfasis en la adaptabilidad ante posibles cambios.

XP fue originalmente diseñada para adecuarse a equipos ya establecidos de 8 a 10 miembros trabajando en proyectos enfocados a la programación orientada a objetos, sin embargo, al pasar el tiempo se ha ido adaptando a equipos más reducidos con el afán de incrementar la productividad. La concepción del desarrollo de XP se popularizó rápidamente entre los desarrolladores de software que se encontraba insatisfechos con metodologías tradicionales como el waterfall.

La documentación debe cumplir con tres criterios: que sea esencial (tenga un nivel mínimo de detalle), sea valiosa (necesaria) y oportuna (realizada justo a tiempo). Inicialmente, se realizan pequeños esfuerzos y los requisitos se centran en capacidades pequeñas y específicas con los clientes y los desarrolladores trabajando en conjunto para trabajar la interacción de los requisitos y las capacidades de un sistema.

Los proyectos de XP implementan ciclos muy cortos con una importante colaboración con los clientes de principio a fin, un trabajo en equipo cerrado, una comunicación abierta y constante entre los participantes, lo que favorece la adaptabilidad de todos los miembros del equipo y el desarrollo incremental.

Prácticas clave en XP:

- Juego de planificación: Se incentiva un trabajo colaborativo entre el cliente y el equipo de desarrollo para estimar y priorizar los requerimientos de cada iteración. Los requerimientos son tomados como historias de usuario y se les asigna secciones y tarjetas correspondientes. Los programadores entonces se dedican a planificar y entregar solo aquellas historias sobre las cuales existe un consenso con el cliente.
- Cambios pequeños: Una versión inicial del sistema se libera a producción después de unas pocas iteraciones. Las características adicionales serán incorporadas en versiones posteriores de acuerdo a la planificación acordada.
- Metáforas: El equipo de desarrollo y el cliente crean un conjunto de metáforas para modelar el sistema que se va a desarrollar.

- **Diseño Simple:** XP incentiva a los desarrolladores para que mantengan el diseño de un sistema tan simple como sea posible. Se trabaja sobre la premisa de que las características de un sistema deberían establecerse una sola vez.
- **Pruebas:** El primer principio aplicado en las pruebas es que los desarrolladores escriben pruebas de aceptación para su código antes de que el código sea propiamente escrito. El cliente determinará las pruebas funcionales a realizarse en cada iteración, de esta manera si se alcanzan las expectativas las pruebas tendrán resultados favorables.
- **Refactorización:** El diseño del sistema puede evolucionar a partir de diseños posteriores de manera que todos los casos de prueba se ejecuten satisfactoriamente.
- **Integración continua:** Todo el código nuevo se integra en el sistema frecuentemente según. Todas las pruebas funcionales deben realizarse después de la integración, de otro modo el nuevo código será descartado.

## **2.2 Herramientas**

En esta sección se presenta una breve revisión de las herramientas utilizadas en el desarrollo del proyecto, a menudo se asume que un proyecto debe tener ciertos componentes funcionales básicos e interdependientes entre sí para poder realizar las funciones requeridas en el sistema, sin embargo, resulta necesario hacer aclaraciones puntuales en la selección de dichas herramientas en contraste con la variedad existente. Saber cuándo utilizar una herramienta por su funcionalidad particular es esencialmente importante en el éxito o el fracaso de un proyecto. En esta sección se detallan las herramientas que se utilizarán en el desarrollo del proyecto.

### **2.2.1 Base de Datos**

Se requiere utilizar una base de datos orientada hacia aplicaciones web, que no genere costos por licenciamiento, tenga buena escalabilidad y desempeño, no requiera de hardware específico y cuya documentación sea detallada y fácilmente accesible.

#### **2.2.1.1 MySQL Community Edition**

MySQL Community Edition es la versión descargable y distribuible de la base de datos de código abierto más popular del mundo. Está disponible bajo la licencia GPL y cuenta con el

respaldo de una enorme y activa comunidad de desarrolladores de código abierto, esta versión incluye:

- Arquitectura de motor de almacenamiento flexible y variada, lo que permite utilizar varios motores de búsqueda y esquemas de almacenamiento como: InnoDB, MyISAM NDB, Memoria, Unir, Archivos, CSV, etc.
- Replicación para mejorar el rendimiento y la escalabilidad de las aplicaciones asociadas a una instancia de MySQL; así como para replicar datos a la vez que proporciona tolerancia a fallas, recuperación failover automática y elasticidad para ofrecer una solución integrada, nativa y orientada a alta disponibilidad.
- Particionamiento para mejorar el rendimiento y la administración de grandes aplicaciones de bases de datos, mejorar la productividad del desarrollador mediante el soporte para creación y aplicación de reglas comerciales complejas a nivel de base de datos.
- Vistas para garantizar que la información confidencial no se vea comprometida. Esquemas de rendimiento para el control de nivel de usuario / aplicación del consumo de recursos y esquemas de información para proporcionar un acceso fácil a los metadatos.

## 2.2.2 Lenguajes de Programación

Se requiere utilizar lenguajes que sean de código abierto, orientados a aplicaciones web, con una sintaxis fácil de comprender, ampliamente documentados e independientes de herramientas específicas como entornos de desarrollo o ejecución determinados.

### 2.2.2.1 PHP

PHP son las siglas recursivas de PHP: Hypertext Preprocessor, se refiere a un lenguaje de código abierto ampliamente utilizado en el mundo del desarrollo web y cuyo principal objetivo es el de renderizar código html en respuesta a determinadas solicitudes por parte de los clientes.

### 2.2.2.2 JavaScript

JavaScript (JS) es un lenguaje de programación ligero que puede ser interpretado o compilado y que implementa funciones de primera clase. Aunque es más conocido como el lenguaje de

secuencias de comandos para las aplicaciones Web, muchos entornos que no son de navegador también lo utilizan, como Node.js, Apache CouchDB y Adobe Acrobat. JavaScript es un lenguaje dinámico de múltiples paradigmas basado en prototipos, que soporta estilos orientados a objetos, imperativos y declarativos (por ejemplo, programación funcional).

#### 2.2.2.3 HTML5

HTML (HyperText Markup Language) es el componente más básico de la Web. Describe y define el contenido de una página web. Otras tecnologías, además de HTML, se utilizan generalmente para describir la apariencia / presentación (CSS) o funcionalidad de una página web (JavaScript).

HTML5 es la última evolución del estándar que define HTML. El término representa dos conceptos diferentes. Es una nueva versión del lenguaje HTML, con nuevos elementos, atributos y comportamientos, y un conjunto más amplio de tecnologías que permite la construcción de sitios web y aplicaciones más diversos y potentes. Este conjunto a veces se denomina HTML5 y amigos y, a menudo, se abrevia solo en HTML5.

#### 2.2.2.4 CSS3

CSS (Cascading Style Sheets) es un lenguaje de hojas de estilo utilizado para describir la presentación de un documento escrito en HTML o XML (incluidos dialectos XML como SVG o XHTML). CSS describe cómo se deben representar los elementos en la pantalla, en papel, en voz o en otros medios.

CSS3 es la última evolución del lenguaje CSS y tiene como objetivo extender CSS2.1. Trae muchas novedades esperadas, como esquinas redondeadas, sombras, degradados, transiciones o animaciones, así como nuevos diseños como columnas múltiples, diseños de caja flexible o cuadrícula.

### 2.2.3 Marcos de Trabajo

Por la naturaleza de los lenguajes utilizados, el marco de trabajo debería tener un buen grado de adaptabilidad para varios entornos de desarrollo, seguir el patrón de arquitectura “Model

Mostrar Controller” (MVC), tener características que faciliten la implementación de políticas de seguridad y ser compatible con el RDBMS.

#### 2.2.3.1 Laravel

Laravel es un marco de trabajo para aplicaciones web que utiliza una sintaxis de PHP expresiva y elegante. Laravel es accesible, potente y proporciona herramientas necesarias para aplicaciones grandes, robustas y seguras. Se basa en que el desarrollo de un producto de software debe ser una experiencia agradable y creativa para ser verdaderamente satisfactoria.

Laravel intenta facilitar algunas tareas comunes de los proyectos web, tales como:

- Enrutamiento.
- Inyección de dependencias.
- Múltiples backend para almacenamiento de sesiones y caché.
- Eloquent, motor ORM expresivo e intuitivo.
- Migraciones de esquemas de base de datos agnósticos.
- Robusto procesamiento de trabajos en segundo plano.
- Publicación de eventos en tiempo real.

Laravel simplifica enormemente la interacción con varios motores de base de datos en una variedad de entornos, se permite usar SQL puro, así como también el fluido constructor de consultas integrado con la herramienta ORM (Object Relation Mapper) Eloquent. Actualmente Laravel soporta cuatro motores por defecto, sin embargo, esta es una característica expandible según se incluyan dependencias al proyecto o cambie la conectividad del backend seleccionado: MySQL, PostgreSQL, SQLite y SQL Server.

La configuración de base de datos para cada aplicación se encuentra detallada en el archivo `database.php` del directorio de configuración. En este archivo se encontrarán definidas las propiedades para todas las conexiones que se requiera realizar, así como la conexión que debería ser usada por defecto.

Requisitos de funcionamiento:

- Versión de PHP equivalente o mayor a 7.0

- Extensión OpenSSL de PHP: vincula las funciones de la librería OpenSSL para cifrado y descifrado simétrico y asimétrico, PBKDF2, PKCS7, PKCS12, X509, etc.
- Extensión PDO PHP: define una interfaz ligera y consistente para acceder a bases de datos usando PHP mediante una capa de abstracción de acceso a datos, lo que significa que independientemente de la base de datos se puede hacer uso de las mismas funciones para realizar consultas y recuperar datos sin reescribir SQL.
- Extensión Mbstring PHP: maneja la conversión de codificación de caracteres entre los posibles pares de codificación y proporciona funciones para manipulación de cadenas específicas de varios bytes que ayudan a manejar las codificaciones multibyte en PHP.
- Extensión Tokenizer PHP: funciones que proporcionan una interfaz para el tokenizer PHP para permitir escribir herramientas de análisis o modificación de fuentes PHP sin tener que lidiar con la especificación del lenguaje a nivel léxico.

#### 2.2.4 Librerías

##### 2.2.3.2 Bootstrap

La capa de presentación debe estar enfocada hacia la maximización de la compatibilidad entre diferentes dispositivos y navegadores, adicionalmente, debe manejar un conjunto determinado de componentes visuales desarrollados en base a un mismo estándar y debería tener una documentación detallada y fácilmente accesible.

Para el caso de estudio se requiere tener la flexibilidad de generar estadísticas personalizadas en los perfiles de los pacientes. Esto significa que se debe incrustar una aplicación en una sección del sistema con el objetivo de procesar datos de forma dinámica, es decir, un script que tome cierto conjunto de datos y pueda generar una representación gráfica de los mismos o cambiar una parte visual de la página para mostrar un resultado.

Bootstrap es un kit de herramientas de código abierto para desarrollar con HTML, CSS y JS. Prototipado rápido de ideas o aplicaciones completas con variables y mixins de SASS, sistema de cuadrícula sensible, componentes precompilados extensos y potentes complementos basados en jQuery. Bootstrap es la librería más popular del mundo para crear sitios responsivos con prioridad de visualización para dispositivos móviles.



## Capítulo 3 – Caso de Estudio

Este capítulo describe los pasos para la generación de los artefactos iniciales que servirán para la aplicación de la metodología de desarrollo seleccionada en función de los requerimientos del caso de estudio. La formalización de requerimientos se realiza en historias de usuario las cuales son el punto de partida necesario para el desarrollo del proyecto y sirven para la generación de la planificación y diseño iniciales.

### 3.1 Historias de Usuario

Para los requerimientos iniciales se puede distinguir cuatro roles principales: paciente, doctor, administrador y asistente; de los cuales solo los roles de doctor y paciente requieren información específica que debe ser gestionada según el esquema de un CRUD. En este sistema cualquier usuario registrado tendrá por defecto el rol de paciente, mientras que los roles de doctor, administrador y asistente deberán ser asignados posteriormente.

El Manifiesto Ágil establece una preferencia por software funcional en lugar de documentación detallada. A diferencia de la definición de requisitos formales, los requisitos ágiles se pueden expresar en breves declaraciones no técnicas llamadas "historias de usuario".

Las historias de usuario son descripciones cortas y simples de una característica contada desde la perspectiva de la persona que utilizará la nueva funcionalidad. Por lo general, siguen una plantilla simple, a continuación, se presentan las cuatro tablas que sintetizan las necesidades del usuario.

Nombre	SD01
Declaración	Como administrador, quiero gestionar las citas, la información de contacto, los médicos, los horarios, la información del paciente, la información de los registros, los roles, la información específica del usuario, realizar copias de seguridad de datos, redactar documentos y también actualizar parámetros específicos.
Pruebas de Aceptación	Como administrador, un usuario debe estar en capacidad de: <ul style="list-style-type: none"><li>• Listar Citas</li></ul>

	<ul style="list-style-type: none"> <li>• Leer Citas</li> <li>• Crear Citas</li> <li>• Borrar Citas</li> <li>• Listar Contactos</li> <li>• Leer Contactos</li> <li>• Editar Contactos</li> <li>• Crear Contactos</li> <li>• Listar Doctores</li> <li>• Mostrar Perfiles de Doctores</li> <li>• Editar Doctores</li> <li>• Crear Doctores</li> <li>• Borrar Doctores</li> <li>• Listar Horarios</li> <li>• Crear Horarios</li> <li>• Borrar Horarios</li> <li>• Listar Pacientes</li> <li>• Mostrar Perfiles de Pacientes</li> <li>• Editar Información de Pacientes</li> <li>• Registrar Usuarios Nuevos</li> <li>• Activar Usuarios</li> <li>• Suspende Usuarios</li> <li>• Listar Registros</li> <li>• Mostrar Registros</li> <li>• Editar Registros</li> <li>• Borrar Registros</li> <li>• Asignar Roles</li> <li>• Revocar Roles</li> <li>• Descargar respaldo de datos</li> </ul>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<ul style="list-style-type: none"> <li>• Actualizar direcciones del catálogo</li> <li>• Redactar documentos</li> </ul>
Tiempo Estimado	8 semanas
Versión	0.2

Tabla 2 Historia de Usuario SD01  
Encalada, C. (junio 2018).

Nombre	SO01
Declaración	Como médico, quiero poder registrar pacientes nuevos, administrar citas, mostrar información de pacientes y otros perfiles de doctores, consultar el historial del paciente, acceder a información de contacto, administrar historiales médicos, redactar certificados y administrar información de pacientes.
Pruebas de Aceptación	<p>Como médico, un usuario debe estar en capacidad de:</p> <ul style="list-style-type: none"> <li>• Listar Citas</li> <li>• Leer Citas</li> <li>• Crear Citas</li> <li>• Borrar Citas</li> <li>• Listar Contactos</li> <li>• Leer Contactos</li> <li>• Editar Contactos</li> <li>• Crear Contactos</li> <li>• Listar Doctores</li> <li>• Mostrar Perfiles</li> <li>• Listar Pacientes</li> <li>• Editar Información de Pacientes</li> <li>• Registrar Usuarios Nuevos</li> <li>• Listar Registros</li> </ul>

	<ul style="list-style-type: none"> <li>• Mostrar Registros</li> <li>• Editar Registros</li> <li>• Borrar Registros</li> <li>• Redactar documentos</li> </ul>
Tiempo Estimado	4 semanas
Versión	0.4

Tabla 3 Historia de Usuario SO01  
Encalada, C. (junio 2018).

Nombre	SG01
Declaración	Como asistente administrativo (es decir, agente), deseo poder registrar Usuarios Nuevos, administrar Citas, Mostrar Información de Pacientes y Perfiles de Doctores, consultar el historial del paciente, redactar documentos a pedido y acceder a la información de contacto.
Pruebas de Aceptación	<p>Como asistente, un usuario debe estar en capacidad de:</p> <ul style="list-style-type: none"> <li>• Listar Citas</li> <li>• Leer Citas</li> <li>• Crear Citas</li> <li>• Borrar Citas</li> <li>• Listar Contactos</li> <li>• Leer Contactos</li> <li>• Editar Contactos</li> <li>• Crear Contactos</li> <li>• Listar Doctores</li> <li>• Mostrar Perfiles de Doctores</li> <li>• Listar Horarios</li> <li>• Listar Pacientes</li> <li>• Mostrar Perfiles</li> </ul>

	<ul style="list-style-type: none"> <li>• Registrar Usuarios Nuevos</li> <li>• Listar Registros</li> <li>• Leer Registros</li> <li>• Redactar documentos</li> </ul>
Tiempo Estimado	2 semanas
Versión	0.6

Tabla 4 Historia de Usuario SG01  
Encalada, C. (junio 2018).

Nombre	SA01
Declaración	Como paciente, quiero poder administrar mis citas, información de contacto y de inicio de sesión, administrar información de mi perfil, listar y mostrar los especialistas disponibles y acceder a mis registros médicos.
Pruebas de Aceptación	<p>Como paciente, un usuario debe estar en capacidad de:</p> <ul style="list-style-type: none"> <li>• Listar Citas</li> <li>• Leer Citas</li> <li>• Crear Citas</li> <li>• Cancelar Citas</li> <li>• Listar Doctores</li> <li>• Mostrar Perfiles de Doctores</li> <li>• Modificar información de cuenta.</li> <li>• Consultar Registros</li> </ul>
Tiempo Estimado	2 semanas
Versión	1

Tabla 5 Historia de Usuario SA01  
Encalada, C. (junio 2018).

Los recursos del sistema representan aquellas clases que pueden ser gestionadas mediante operaciones de Creación, Listado, Lectura, Actualización o Eliminación (BREAD). Según la historia de usuario inicial, los recursos del sistema son: usuarios, registros, signos vitales, citas, horarios, archivos y datos de contacto.

### 3.2 Planificación

Los requerimientos definen las capacidades que un sistema debe tener (funcional) o las propiedades de ese sistema (no funcional) que satisfacen las necesidades de los usuarios para realizar un conjunto específico de tareas (dentro de un alcance definido).

Las metodologías ágiles comúnmente desarrollan un sistema en el transcurso de iteraciones sucesivas (sprints) utilizando las historias de usuario como entradas planificadas para los sprints. Documentar los requisitos es parte del proceso de desarrollo y se enfoca en lo que realmente se ha hecho.

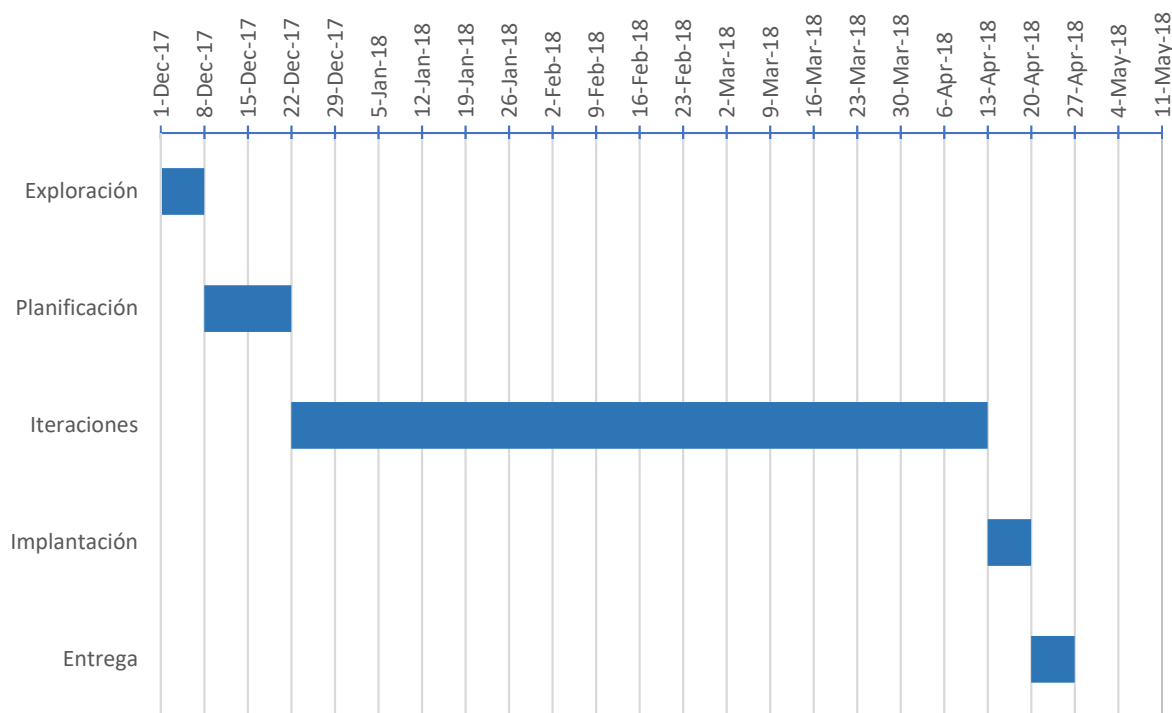


Figura 1 Planificación Inicial  
Encalada, C. (junio 2018).

### 3.2.1 Información complementaria

Las clases correspondientes a información complementaria están asociadas a las tablas que contienen la información relevante para los usuarios según el rol que desempeñen en el sistema: roles, pacientes y doctores. En el caso particular de los administradores del sistema se asume que no tienen información específica sobre sí mismos salvo por sus datos de usuario.

### 3.2.2 Casos de Uso

Un diagrama de casos de uso es un diagrama dinámico o de comportamiento en UML. Los diagramas de casos de uso modelan la funcionalidad de un sistema usando actores y capacidades de un sistema. Los casos de uso son un conjunto de acciones, servicios y funciones que el sistema necesita realizar. En este contexto, un "sistema" es algo que se está desarrollando u operando, como un sitio web. Los "actores" son personas o entidades que operan bajo roles definidos dentro del sistema.

#### 3.2.3.1 Nivel General

Un caso de uso a nivel general detallará el conjunto de acciones o pasos relacionados a eventos que definen las interacciones entre todos los actores, sin tomar en cuenta su rol asociado, y el sistema para lograr un objetivo.

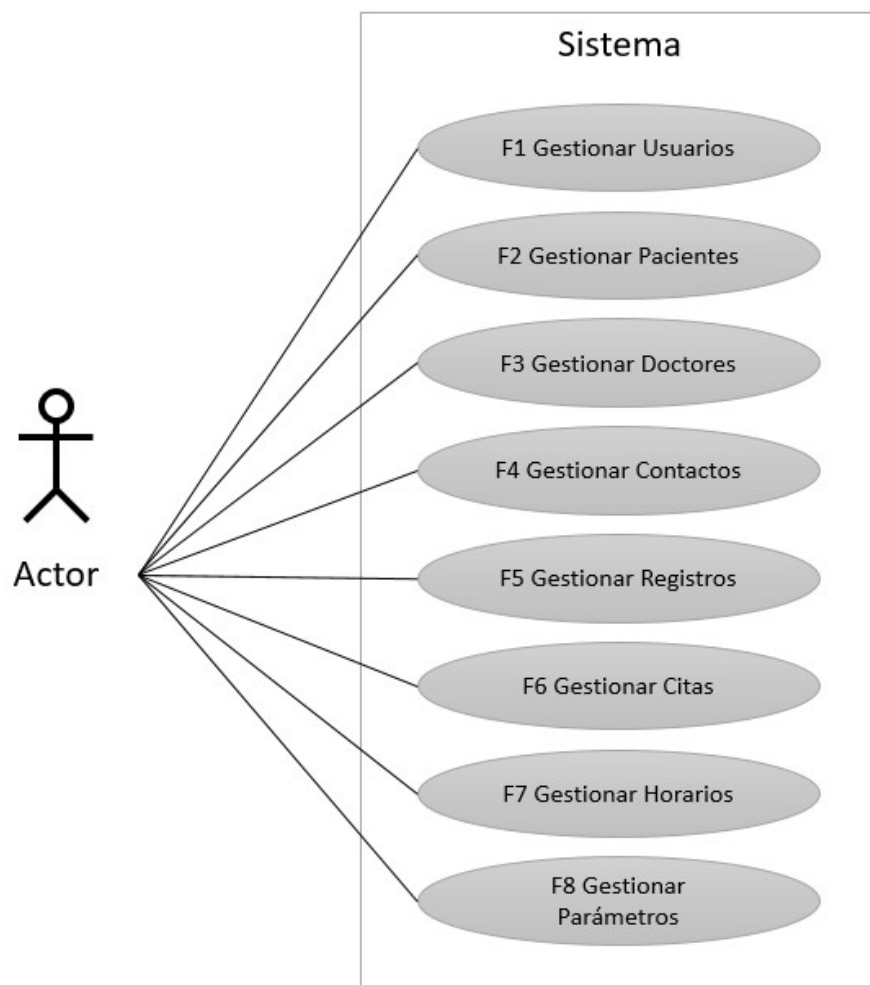


Figura 2 Diagrama general de casos de uso  
Encalada, C. (junio 2018).

### 3.2.3.2 Siguiente Nivel

En este nivel de detalle para casos de uso, el actor puede ser un ser humano u otro sistema externo, tomando en cuenta el rol que desempeña el mismo para detallar las interacciones desde un punto de vista más específico.



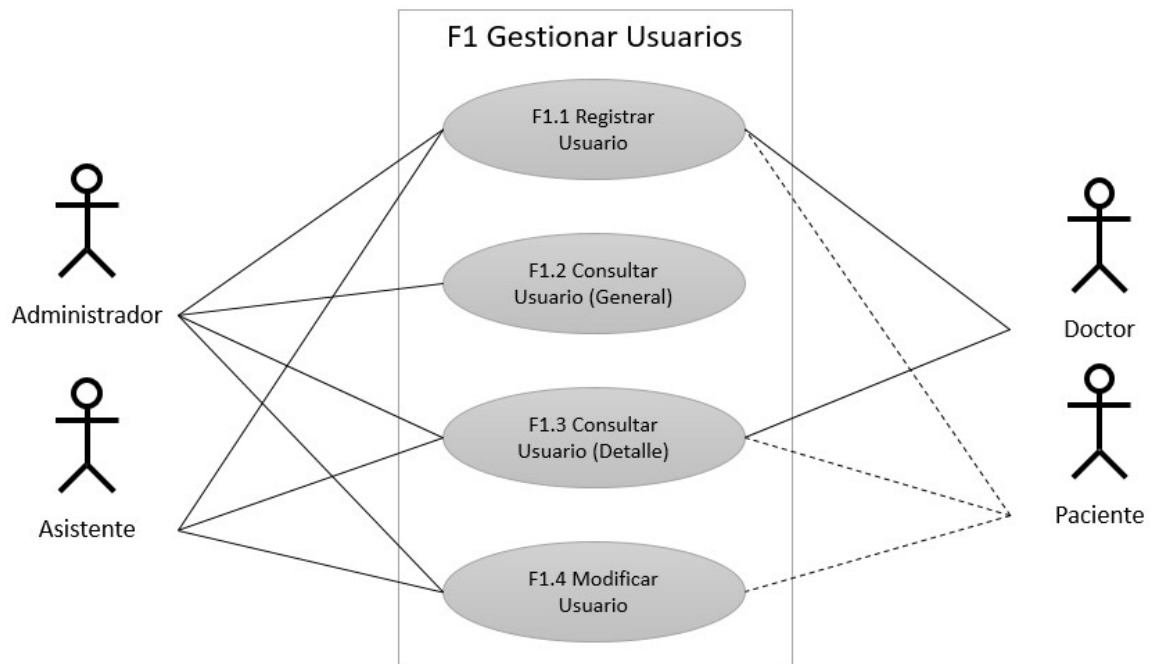


Figura 3 Gestión de usuarios  
Encalada, C. (junio 2018).

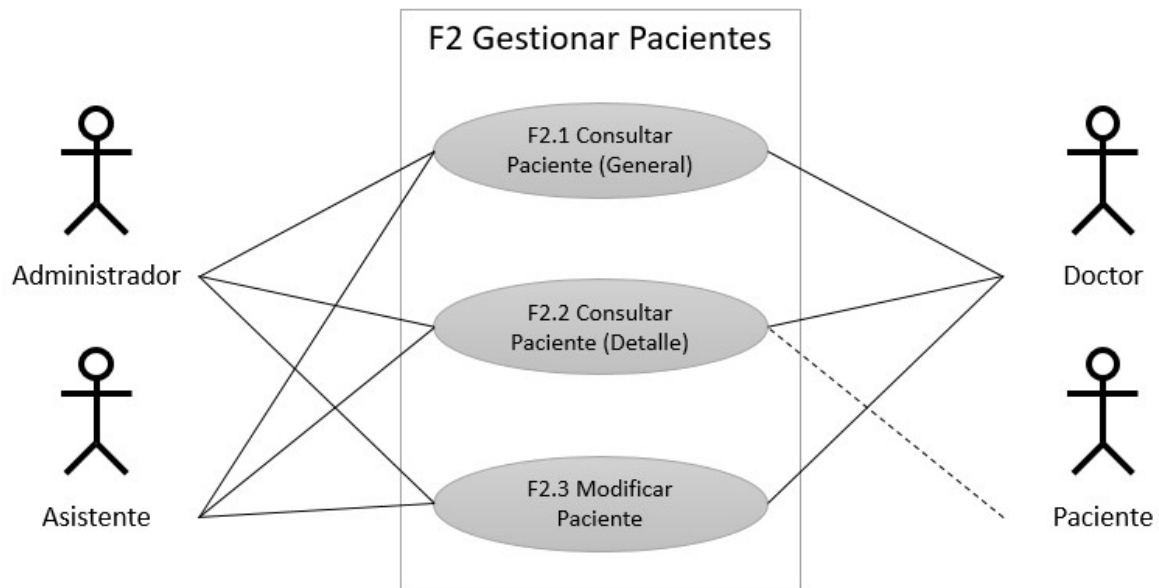


Figura 4 Gestión de pacientes  
Encalada, C. (junio 2018).

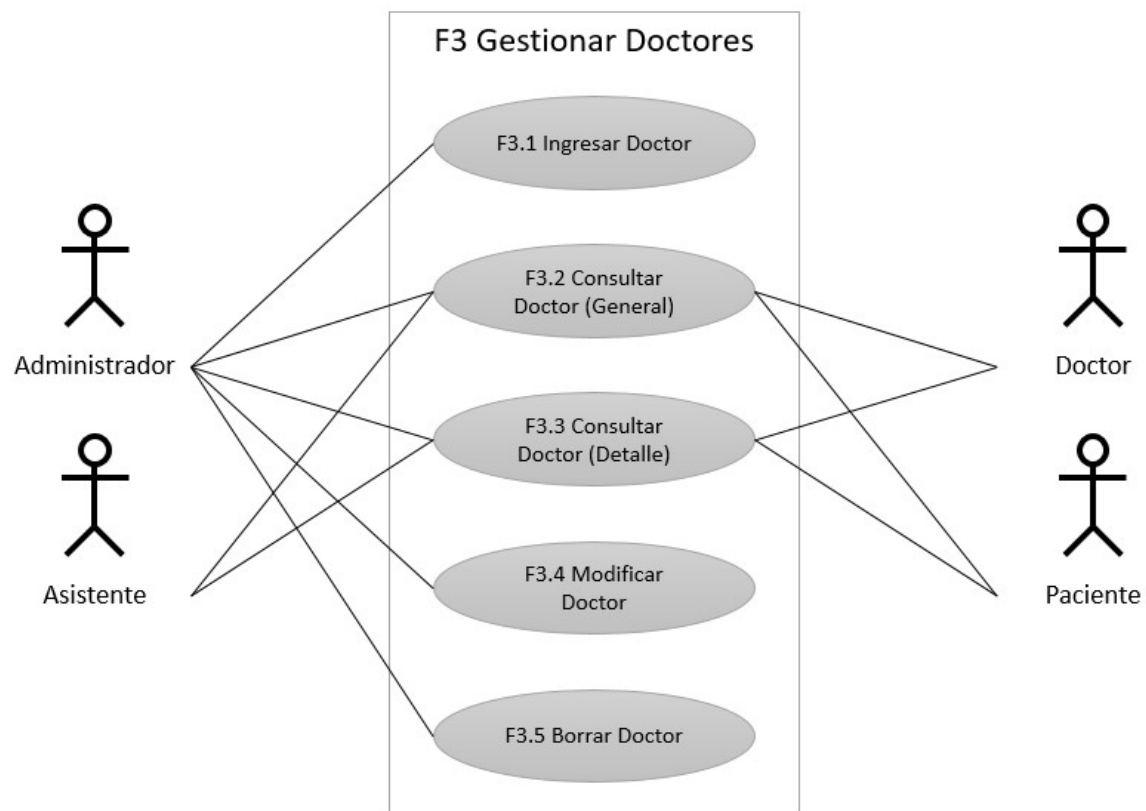


Figura 5 Gestión de doctores  
Encalada, C. (junio 2018).

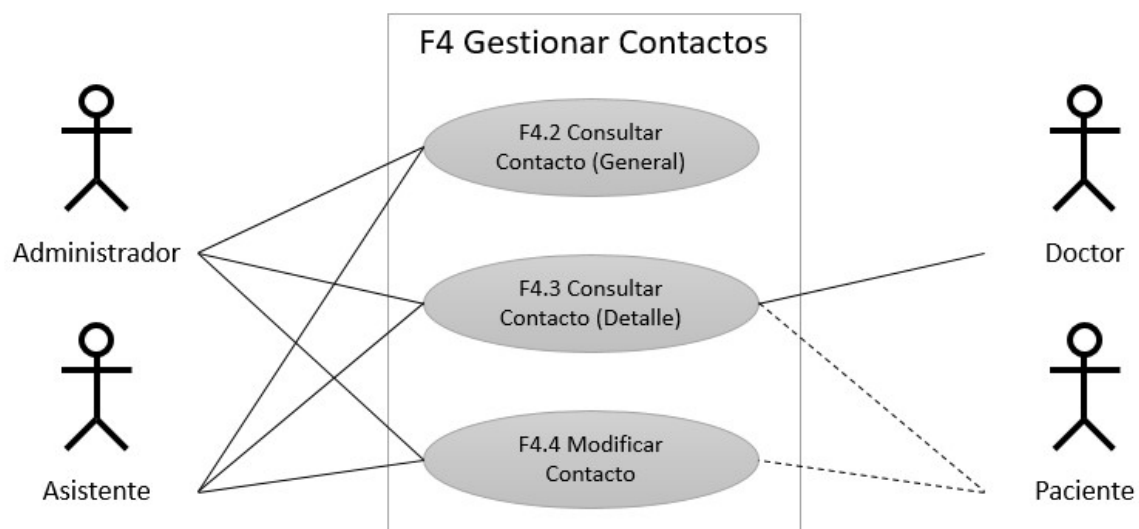


Figura 6 Gestión de contactos  
Encalada, C. (junio 2018).

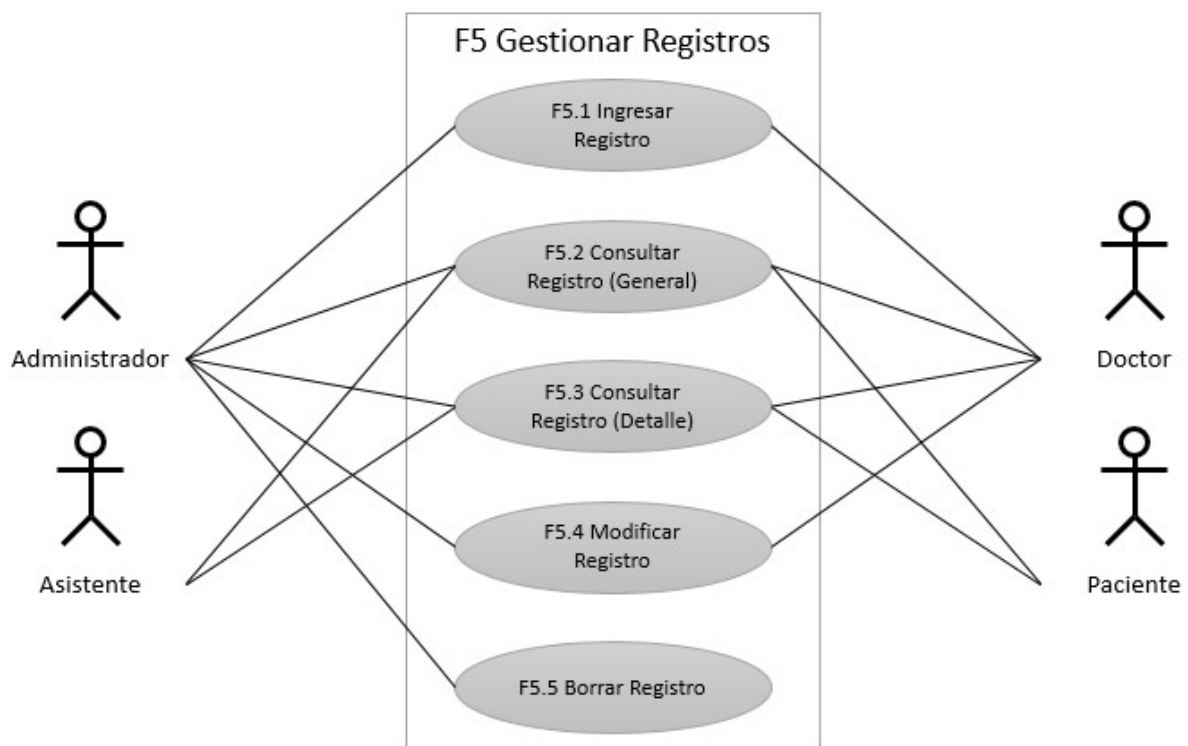


Figura 7 Gestión de registros  
Encalada, C. (junio 2018).

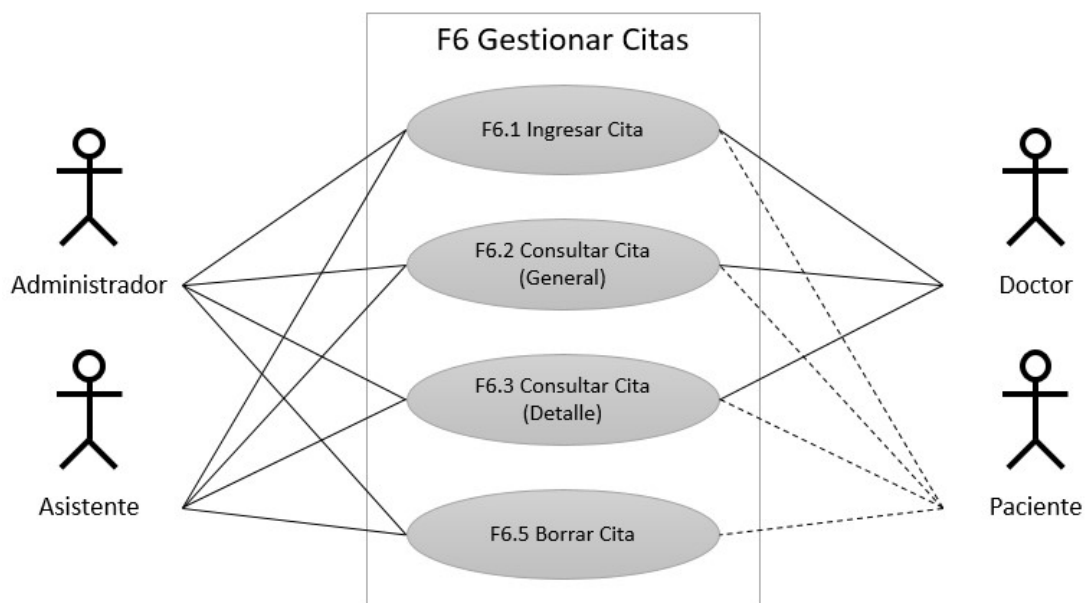


Figura 8 Gestión de citas médicas.  
Encalada, C. (junio 2018).

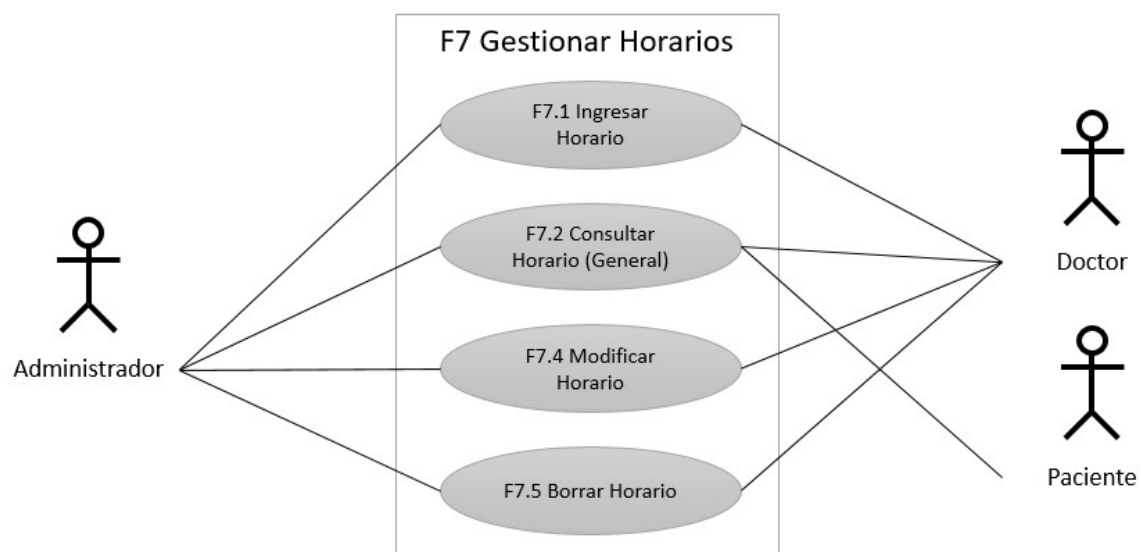


Figura 9 Gestión de horarios de doctores.  
Encalada, C. (junio 2018).

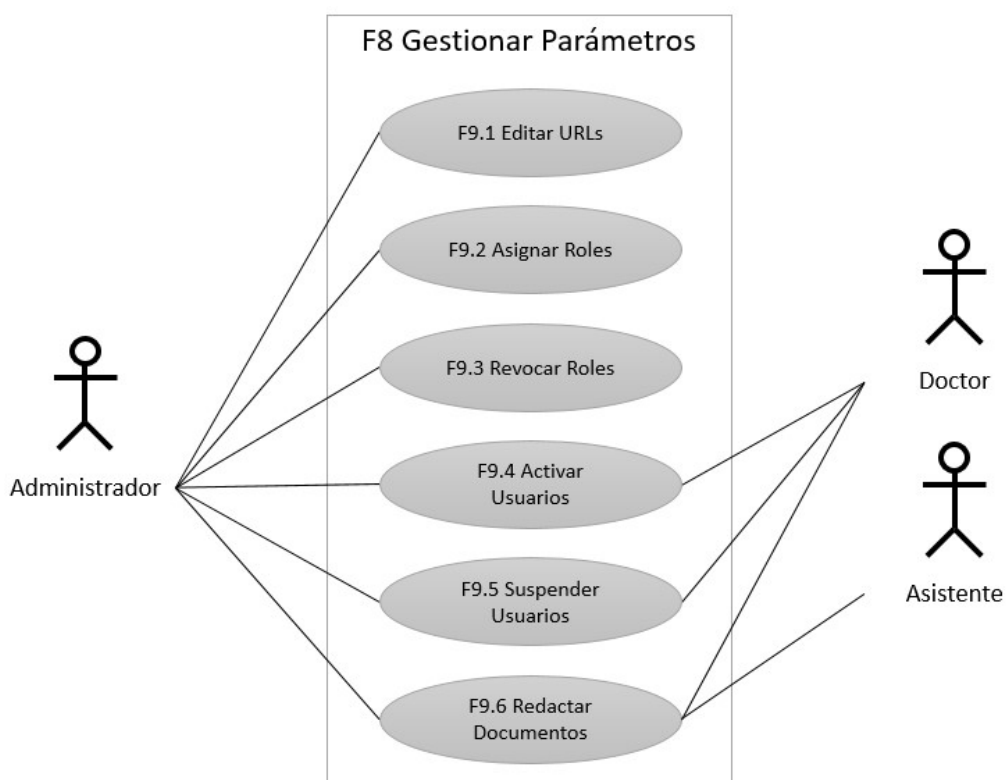


Figura 10 Gestión de parámetros.  
Encalada, C. (junio 2018).

### 3.2.3 Formalización de Requerimientos

En esta sección se detallan los requerimientos funcionales y no funcionales que corresponden respectivamente a la funcionalidad del sistema y como debería operar el mismo.

Requerimiento Funcional	Requerimiento de Diseño
Se requiere llevar el historial de al menos 400 pacientes.	<ul style="list-style-type: none"><li>• Base de datos.</li><li>• Gestión de Usuarios.<ul style="list-style-type: none"><li>○ Gestión de Roles con información específica.</li></ul></li></ul>
Se requiere manejar información de pediatría para realizar cálculos adicionales.	<ul style="list-style-type: none"><li>• Gestión de Historiales.</li></ul>
Se requiere un módulo de gestión de archivos auxiliares.	<ul style="list-style-type: none"><li>• Gestión de Archivos.</li></ul>
Se requiere un módulo de reservación y consulta de citas médicas.	<ul style="list-style-type: none"><li>• Gestión de Citas.</li></ul>

Tabla 6 Requerimientos funcionales  
Encalada, C. (junio 2018).

Requerimientos no Funcionales	Detalle
Desempeño	<ul style="list-style-type: none"><li>• El sistema debe ser capaz de responder a todas las solicitudes que recibe.</li></ul>
Seguridad	<ul style="list-style-type: none"><li>• El sistema debe mostrar la información pertinente.</li><li>• Todos los usuarios del sistema deben ser autenticados.</li><li>• La base de datos debe aplicar reglas para mantener integridad de datos.</li></ul>
Confidencialidad	<ul style="list-style-type: none"><li>• Los datos de los pacientes deben estar seguros.</li></ul>

	<ul style="list-style-type: none"> <li>• Los datos de los pacientes no deberán ser compartidos con otros sistemas.</li> </ul>
Pertinencia	<ul style="list-style-type: none"> <li>• El sistema debe validar la integridad de los datos ingresados.</li> <li>• El sistema debe presentar información confiable.</li> </ul>

Tabla 7 Requerimientos no funcionales  
Encalada, C. (junio 2018).

Requerimientos Mínimos	Detalle
Hardware de Cliente	<ul style="list-style-type: none"> <li>• No aplica</li> </ul>
Software de Cliente	<ul style="list-style-type: none"> <li>• Navegador Web que soporte para JavaScript y HTML5</li> </ul>

Tabla 8 Requerimientos mínimos  
Encalada, C. (junio 2018).

### 3.3 Diseño

En este capítulo se presenta el diseño, la estructura conceptual y la arquitectura por módulos que se aplicará, la intención específicamente es detallar la estructura modular del proyecto de software de forma efectiva.

El criterio de éxito del proyecto dependerá de cuan bien se acople la implementación final al diseño propuesto inicialmente, es decir, si existieran cambios durante el desarrollo del proyecto se debe procurar no alterar fundamentalmente las estructuras propuestas al inicio sino, construir sobre éstas para satisfacer las necesidades del cliente.

### 3.3.1 Modelo Conceptual

El modelo conceptual detalla de forma abstracta las asociaciones que existen entre las diferentes entidades que componen la capa de datos de un sistema.

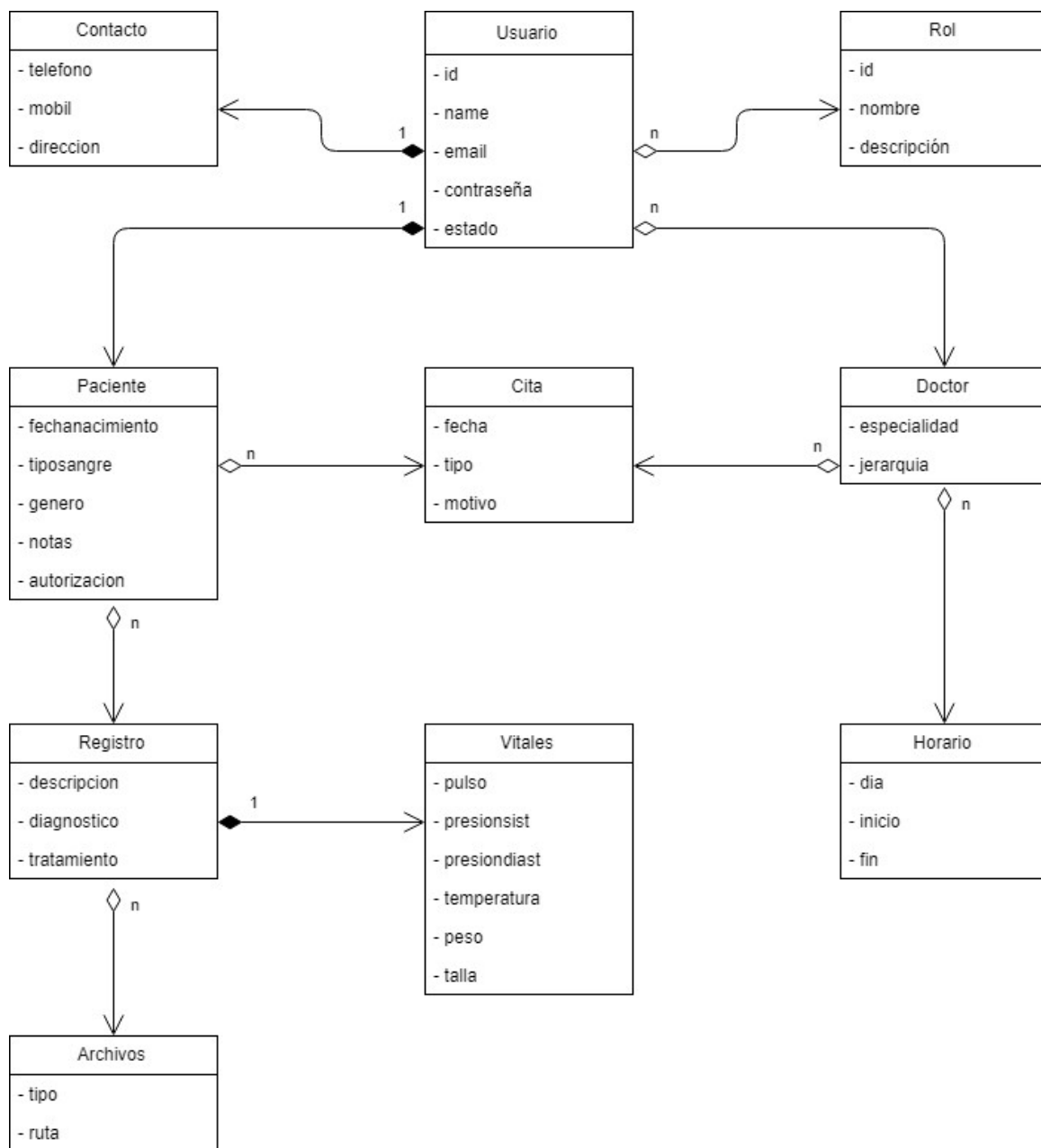


Figura 11 Diseño conceptual.  
Encalada, C. (junio 2018).

### 3.3.2 Estructura de Base de Datos

La estructura de base de datos corresponde al llamado diseño físico de datos, efectivamente representando mediante tablas el esquema detallado anteriormente como modelo conceptual, en este punto se identifican claramente las relaciones con atributos que vinculan los registros existentes en distintas tablas.

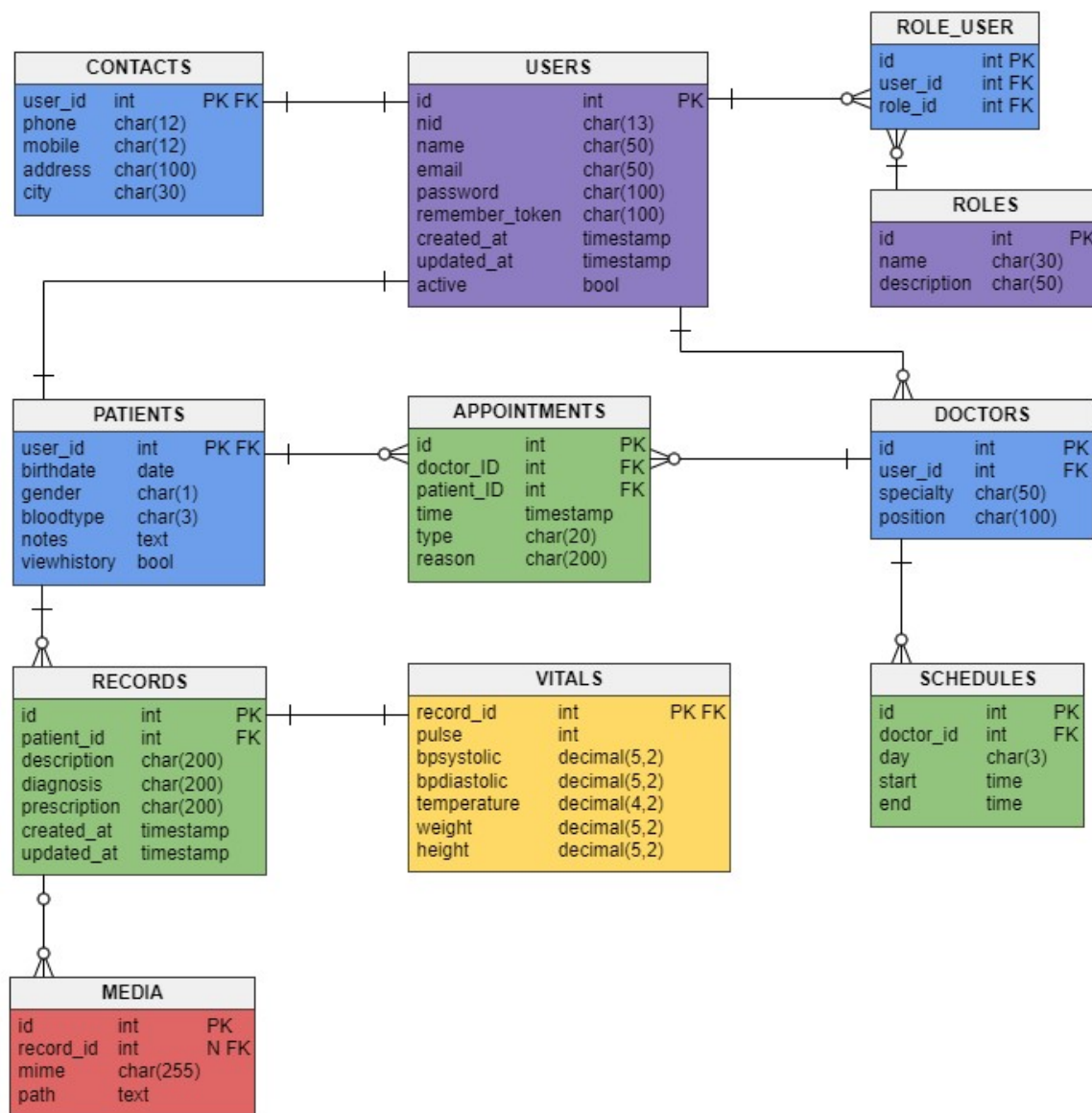


Figura 12 Diseño de base de datos.  
Encalada, C. (junio 2018).





### 3.4.1 Artisan Console

Artisan es la interfaz de línea de comandos incluida con Laravel, proporciona una serie de útiles comandos que ayudan en la construcción de la aplicación.

### 3.4.2 Migraciones

Las migraciones son control de versionamiento para la base de datos, esto permite al equipo de desarrollo modificar y compartir fácilmente el esquema de la base de datos de la aplicación. Las migraciones suelen combinarse con el generador de esquemas de Laravel para crear fácilmente el esquema de la base de datos de una aplicación.

Esta forma de trabajo elimina la necesidad de solicitar cambios manualmente a una columna o estructura de base de datos. La interfaz “Schema” de Laravel proporciona el soporte para la creación y manipulación de tablas en todos los sistemas de bases de datos admitidos de forma transparente, maximizando de esta manera la compatibilidad y portabilidad de la aplicación.

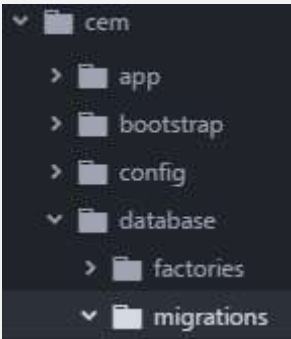
Ruta	Detalle
	<p>Orden de llamada:</p> <ol style="list-style-type: none"><li>1. Users, Roles.</li><li>2. Role_User, Contactos, Pacientes, Doctores.</li><li>3. Citas, Horarios, Records.</li><li>4. Vitals.</li><li>5. Media.</li></ol> <p>Resultado esperado: creación de estructura de base de datos.</p>

Tabla 9 Detalle de migraciones.  
Encalada, C. (junio 2018).

### 3.4.3 Datos de Prueba

Laravel incluye un método simple para llenar una base de datos con datos de prueba utilizando las clases “seed” (semilla). Todas las clases semilla se almacenan en el directorio de database/seeds.

Las clases semilla pueden tener cualquier nombre según sea su propósito, sin embargo, se recomienda seguir alguna convención sensata como PatientTableSeeder, etc. De forma predeterminada, se define una clase de DatabaseSeeder, desde esta clase se puede invocar el método para ejecutar otras clases semilla, lo que le permite controlar la orden de inicialización.

Especificificar manualmente los atributos para cada semilla de modelo es un proceso largo, en su lugar, se puede usar fábricas de modelo para generar convenientemente grandes cantidades de registros de bases de datos; primero se revisa la estructura de los modelos para luego poder definir sus correspondientes fábricas, una vez hecho esto se invoca la función de fábrica “factory” de cada modelo para insertar registros en la base de datos.

Ruta	Detalle
	 <pre> use Illuminate\Database\Seeder;  class DatabaseSeeder extends Seeder {     public function run()     {         \$this-&gt;call(RoleSeeder::class);         \$this-&gt;call(UserSeeder::class);         \$this-&gt;call(ContactSeeder::class);         \$this-&gt;call(PatientSeeder::class);         \$this-&gt;call(DoctorSeeder::class);         \$this-&gt;call(AppointmentSeeder::class);         \$this-&gt;call(RecordSeeder::class);         \$this-&gt;call(ScheduleSeeder::class);         \$this-&gt;call(VitalSeeder::class);         \$this-&gt;call(MediumSeeder::class);     } } </pre> <p>Los seeder son llamados en el orden en que se muestra.</p>

Tabla 10 Generación de datos de prueba.  
Encalada, C. (junio 2018).

### 3.4.5 Esquema de Interfaces

Todas las Interfaces gráficas fueron diseñadas en base a componentes reutilizables especificados en Twitter Bootstrap v4.0. A continuación se detalla el esquema general sobre el que se basará la interfaz de usuario:

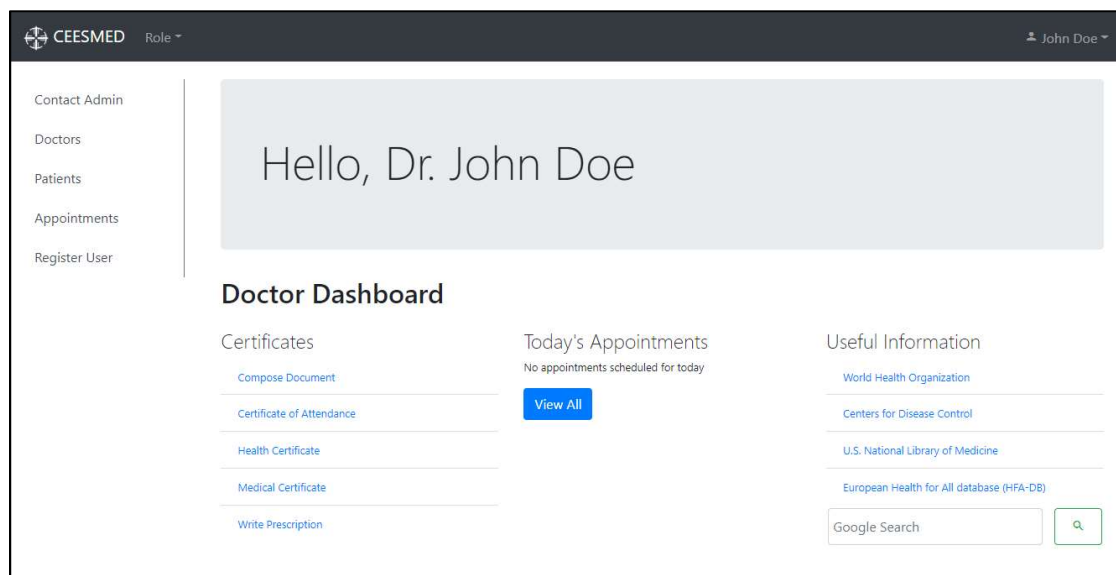


Figura 14 Esquema de interfaz gráfica.  
Encalada, C. (junio 2018).

Se puede apreciar la aplicación de la navegación, la barra lateral y el área de contenido principal. En la navegación se tiene las interacciones que corresponden a las acciones que tienen que ver con la cuenta del usuario como actor individual. En la barra lateral se tendrán las acciones permitidas para cada usuario según el rol que tenga asignado. El área principal o de contenido mostrará la información y componentes pertinentes para que el usuario pueda interactuar con la información del sistema.

### 3.5 Pruebas

La tabla a continuación detalla las pruebas unitarias realizadas sobre la primera iteración y su resultado respecto al rol que las llevó a cabo.

Método	URI	Nombre	Paciente	Asistente	Doctor	Administrador
GET	contact	externals.contact	OK	OK	OK	OK
GET	locate	externals.locate	OK	OK	OK	OK
GET	/	externals.welcome	OK	OK	OK	OK
GET	home/{d?}	home	OK	OK	OK	OK
GET	login	login	OK	OK	OK	OK
POST	logout	logout	OK	OK	OK	OK
POST	password/email	password.email	OK	OK	OK	OK
GET	password/reset	password.request	OK	OK	OK	OK
GET	register	register	OK	OK	OK	OK
POST	login		OK	OK	OK	OK
POST	register		OK	OK	OK	OK

Tabla 11 Pruebas unitarias iniciales.  
Encalada, C. (junio 2018).

## Capítulo 4 – Otras Iteraciones

En este capítulo se detallan los pasos siguientes tomados en las siguientes iteraciones en concordancia con los objetivos del proyecto e historias de usuario.

### 4.1 Versiones

A continuación, se detalla el esquema sobre el cual se fue desarrollando el proyecto.

Versión	Historia	Características									
		Citas	Contactos	Doctores	Horarios	Pacientes	Registros	Perfiles	Configuración		
0.2	SD01 Administrador	Listar	Listar	Listar	Listar	Listar	Listar	Ver Paciente	Registrar Usuario		
		Mostrar	Mostrar				Mostrar	Ver Doctor	Suspender Usuario		
			Editar	Editar		Editar	Editar		Activar Usuario		
		Crear		Crear	Crear		Crear		Asignar Rol		
		Borrar		Borrar	Borrar		Borrar		Revocar Rol		
									Respaldar Datos		
									Actualizar URLs		
									Redactar Documentos		
0.4	SO01 Doctor	Listar	Listar	Listar	Listar	Listar	Listar	Ver Paciente	Registrar Usuario		
		Mostrar	Mostrar			Editar	Mostrar	Ver Doctor	Redactar Documentos		
			Editar				Editar				
		Crear					Crear				
		Borrar					Borrar				

Figura 15 Esquema de iteraciones (pt. 1)  
Encalada, C. (junio 2018).



Patient Index					<input type="text" value="Name or Email"/> <input type="button" value="Search"/>
Name	Date of birth	Gender	History Enabled	Bloodtype	
Abdullah Hodkiewicz	February 25, 2014	female	no	*	<a href="#">Profile</a> <a href="#">Update</a>
Carlos Encalada	August 22, 1990	male	no	O -	<a href="#">Profile</a> <a href="#">Update</a>
Celestine Harvey PhD	May 28, 2011	female	no	O +	<a href="#">Profile</a> <a href="#">Update</a>
Deborah Turcotte	October 13, 2008	female	no	O *	<a href="#">Profile</a> <a href="#">Update</a>
Dell Collins PhD	September 20, 2013	male	no	O -	<a href="#">Profile</a> <a href="#">Update</a>
Dr. Jillian Connelly	January 10, 2013	male	no	O *	<a href="#">Profile</a> <a href="#">Update</a>
Juan Perez	April 20, 2000	male	no	*	<a href="#">Profile</a> <a href="#">Update</a>
Lenin Villacres	June 22, 1950	male	no	O +	<a href="#">Profile</a> <a href="#">Update</a>
Lenin Villacres	June 7, 2018	male	no	*	<a href="#">Profile</a> <a href="#">Update</a>
Marshall Nicolas	February 6, 2013	female	no	A *	<a href="#">Profile</a> <a href="#">Update</a>
Max Carroll	December 31, 2014	undisclosed	no	AB-	<a href="#">Profile</a> <a href="#">Update</a>
Miss Germaine Waters Jr.	January 23, 2014	female	no	O -	<a href="#">Profile</a> <a href="#">Update</a>

Figura 17 Prototipo de indexado.  
Encalada, C. (junio 2018).


 <p>Change Avatar</p> <p><b>Celestine Harvey PhD</b></p> <hr/> <p><a href="#">Generate Appointment</a></p> <p><a href="#">View Appointments</a></p> <p><a href="#">Generate Record</a></p> <p><a href="#">Medical History</a></p> <p><a href="#">Doctor Profile</a></p> <p><a href="#">Password Reset</a></p> <p><a href="#">Activate</a></p>	<b>Patient Information</b> <a href="#">Update</a>	<b>User Data</b> <a href="#">Update</a>		
	Age	7 years, 0 months	ID	1499381069
	Gender	female	E-Mail	monica.dubuque@example.net
	Bloodtype	O +		
	<b>Latest history</b>		<b>Contact information</b> <a href="#">Update</a>	
	Anamnesis	Voluptas accusantium qui quo quam excepturi vel quos.		
	Diagnosis	Qui tempore blanditiis beatae quia saepe. Eum tenetur quibusdam vero ut laborum harum. Iure reiciendis et sit occaecati fugit in voluptas. Quo molestiae alias voluptate unde et quia. Placeat voluptates molestiae illum velit. Ipsam sunt dolores sapiente culpa voluptatem.		
	Prescription	Ut unde sunt omnis exercitationem. Dicta exercitationem consequatur voluptatem. Et quaerat ullam tenetur eos qui aliquam consequatur. Iste nisi mollitia dolorem quos.		
	Pulse	77 bpm	Phone	920835602
	Temperature	36.99 °C	Address	6280 Jarrod Views Apt. 189 West Nick, NE 03418
Systolic p.	131.87 mm Hg	Mobile	7858095108	
Diastolic p.	91.06 mm Hg	City	Port Tyrique	
Height	55.21 cm			
Weight	74.17 kg			

Figura 18 Prototipo de perfil.  
Encalada, C. (junio 2018).



### New Patient

Name	ID
<input type="text"/>	<input type="text"/>
Email	Date of birth
<input type="text" value="patient@ceesmed.com"/>	<input type="text" value="mm/dd/yyyy"/>
Gender	Bloodtype
<input type="text"/>	<input type="text"/>
Background	
<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"> Pathological History:   Alergies:   Vaccines:   Other: </div>	
Phone	Mobile
<input type="text"/>	<input type="text"/>
Address	City
<input type="text"/>	<input type="text" value="Quito"/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

Figura 19 Prototipo de creación y edición.  
Encalada, C. (junio 2018).

### 4.3 Pruebas de Aceptación

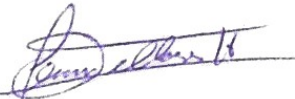
Las pruebas de aceptación constituyen las listas de verificación que evidencian el cumplimiento de los requerimientos del sistema desde la perspectiva del usuario, los documentos de aceptación correspondientes a las iteraciones realizadas se encuentran a continuación, el documento de aceptación final correspondiente a la versión 1.0 del producto se encuentra al final de este documento en la sección de anexos.


## Aceptación V 0.2

Quito, 12 de enero de 2018

Yo, Lenin Villacrés, en calidad de Director del Centro de Especialidades médicas declaro que el sistema presenta una funcionalidad comprobada para las tareas a continuación detalladas:

<input checked="" type="checkbox"/>	Gestionar Citas
<input checked="" type="checkbox"/>	Gestionar Contactos
<input checked="" type="checkbox"/>	Gestionar Doctores
<input checked="" type="checkbox"/>	Gestionar Horarios
<input checked="" type="checkbox"/>	Gestionar Pacientes
<input checked="" type="checkbox"/>	Gestionar Registros
<input checked="" type="checkbox"/>	Gestionar Roles
<input checked="" type="checkbox"/>	Gestionar Configuración

  
Dr. Lenin Villacrés

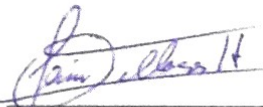
  
Carlos Encalada


## Aceptación V 0.4

Quito, 2 de Febrero de 2018

Yo, Lenin Villacrés, en calidad de Director del Centro de Especialidades médicas declaro que el sistema presenta una funcionalidad comprobada para las tareas a continuación detalladas:

- |                                     |                     |
|-------------------------------------|---------------------|
| <input checked="" type="checkbox"/> | Gestionar Citas     |
| <input checked="" type="checkbox"/> | Gestionar Contactos |
| <input checked="" type="checkbox"/> | Gestionar Pacientes |
| <input checked="" type="checkbox"/> | Gestionar Registros |

  
\_\_\_\_\_  
Dr. Lenin Villacrés

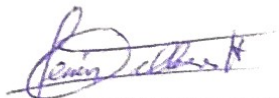
  
\_\_\_\_\_  
Carlos Encalada


## Aceptación V 0.6

Quito, 23 de Febrero de 2018

Yo, Lenin Villacrés, en calidad de Director del Centro de Especialidades médicas declaro que el sistema presenta una funcionalidad comprobada para las tareas a continuación detalladas:

- |                                     |                     |
|-------------------------------------|---------------------|
| <input checked="" type="checkbox"/> | Gestionar Citas     |
| <input checked="" type="checkbox"/> | Gestionar Contactos |
| <input checked="" type="checkbox"/> | Gestionar Pacientes |

  
\_\_\_\_\_  
Dr. Lenin Villacrés

  
\_\_\_\_\_  
Carlos Encalada

## 4.4 Rutas

Las rutas definen como se accederá a los recursos del sistema, los servicios web y las vistas específicas. No se ha determinado hasta el momento la necesidad de utilizar servicios web, por este motivo el archivo `api.php` se encuentra vacío.

No se ha determinado la necesidad de definir comandos adicionales para la consola Artisan, por este motivo el archivo `console.php` se encuentra con las definiciones por defecto.

No se ha determinado hasta el momento la necesidad de definir eventos a los cuales se pueda suscribir clientes, por este motivo el archivo `channels.php` se encuentra con la definición por defecto, a continuación, se presentan las tablas que corresponden a las rutas de acceso de la aplicación.

Tipo	URI	Método	Nombre
get		HomeController@index	externals.welcome
get	/	HomeController@index	externals.welcome
get	locate	HomeController@locate	externals.locate
get	contact	HomeController@contact	externals.contact
post	contact	HomeController@reach	externals.reach

Tabla 12 Rutas no protegidas.  
Encalada, C. (junio 2018).

En la tabla 12 se encuentran las rutas de acceso que no están protegidas por autenticación de usuario, es decir, aquellas que son públicamente accesibles.

Tipo	URI	Controlador	Filtro
resource	media	MediaController	[only => [destroy, show]]
resource	users	UserController	[except => [store]]
resource	doctors	DoctorController	[except => [show]]
resource	appointments	AppointmentController	[except => [edit, update]]
resource	patients	PatientController	[except => [show, destroy]]
resource	schedules	ScheduleController	[except => [show, edit, update]]
resource	roles	RoleController	[except => [show, edit, update]]
resource	records	RecordController	[except => [index, show, create]]
resource	contacts	ContactController	[except => [show, create, store, destroy]]

Tabla 13 Recursos, Controladores y Filtros.  
Encalada, C. (junio 2018).

En la tabla 13 se encuentran las rutas de recurso que corresponden a los registros que son gestionados por la aplicación, los controladores son las clases que contiene la lógica necesaria

para realizar operaciones sobre los registros o para mostrar información; en Laravel se tiene la posibilidad de incluir o excluir ciertas rutas pre-configuradas mediante la utilización de filtros.

Tipo	URI	Método	Nombre
Get	home/{d?}	HomeController@index	home
Get	backup	DatabaseController@backup	databases.backup
Get	getbackup	DatabaseController@getBackup	databases.download
Get	users/{user}/avatar	UserController@editAvatar	avatar.edit
Post	avatar/{user}	UserController@updateAvatar	avatar.update
Get	write/{did?}	HomeController@write	home.write
Post	print	HomeController@print	home.print
Get	catalogs	HomeController@editCatalog	catalogs.edit
Put	catalogs	HomeController@updateCatalog	catalogs.update
Get	vitals/{pid}	VitalController@index	vitals.index
Get	file/{medium}	MediaController@deliver	media.deliver
get	schedules/{did}	ScheduleController@show	schedules.show
get	schedules/{did}/{json?}	ScheduleController@show	schedules.list
get	doctors/{pid?}/{did?}	DoctorController@index	doctors.index
get	users/{user}/suspend	UserController@suspend	users.suspend
post	users/{user}/activate	UserController@activate	users.activate
get	doctor/{id?}	UserController@profile	profile.doctor
get	patient/{id?}	UserController@profile	profile.patient
get	patients/search/{term}	PatientController@find	patients.find
post	patients/search	PatientController@search	patients.search
get	myrecords	RecordController@myRecords	records.my
get	mydoctors	DoctorController@myDoctors	doctors.my
get	changepassword	UserController@changePassword	password.change
post	changepassword	UserController@updatePassword	password.update
get	myappointments	AppointmentController@myAppointments	appointments.my
get	records/{pid}	RecordController@index	records.index
get	records/create/{pid}	RecordController@create	records.create
get	records/{pid}/{record}	RecordController@show	records.show
get	todaycontacts	ContactController@today	contacts.today
get	appointments/today/{id?}	AppointmentController@today	appointments.today
get	appointments/{pid?}/{did?}	AppointmentController@index	appointments.index
get	appointments/create/{pid?}/{did?}	AppointmentController@create	appointments.create

Tabla 14 Rutas específicas.  
Encalada, C. (junio 2018).

En la tabla 14 se detallan las rutas creadas para satisfacer los requerimientos del proyecto, el controlador responsable de gestionar la ruta, el método utilizado separado por el carácter “@” y finalmente el nombre de la ruta asociada.

## 4.5 Controladores

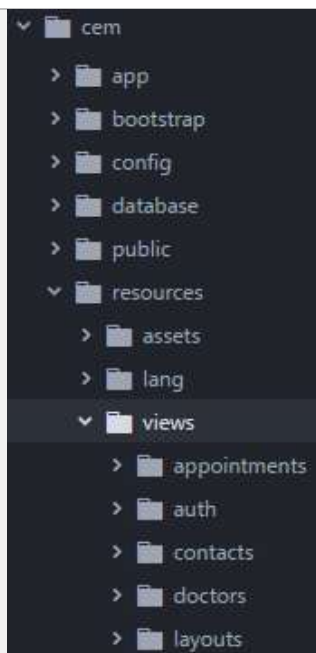
En lugar de definir toda la lógica de gestión de solicitudes como cierres en archivos de ruta o archivos independientes, se organiza estos procesos utilizando clases de controlador. Los controladores pueden agrupar la lógica de manejo de solicitudes relacionada en una sola clase. Los controladores se almacenan en el directorio App/Http/Controllers.

El enrutamiento de recursos de Laravel asigna las rutas "CRUD" típicas a un controlador con una sola línea de código. Por ejemplo, es posible que crear un controlador que maneje todas las solicitudes HTTP para el recurso "horarios" de la aplicación.

## 4.6 Vistas

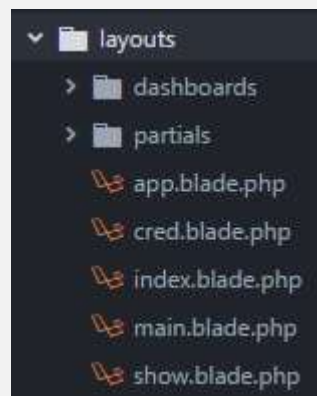
Las vistas corresponden a los archivos que definen la capa de presentación según la interacción que se vaya a realizar y el usuario que desea realizarla, las vistas están basadas en el motor de generación de plantillas Blade.

Archivos	Detalle
	<p>Los directorios appointments, contacts, doctors, media, patients, records, schedules y vitals; contienen las vistas personalizadas que permiten realizar las funciones de CRUD según el recurso particular al que se refieran. Las mencionadas vistas heredarán su estructura principal de la vista principal main.blade.php.</p> <p>El directorio auth contiene todas las vistas correspondientes a las posibles interacciones del usuario con respecto a su cuenta.</p>



El directorio vendor contiene las vistas que son utilizadas en Laravel para funciones específicas como la paginación o renderización de contenido complementario.

El directorio layouts contiene los archivos base sobre los cuales se realiza la resolución de plantillas del resto de archivos:



El archivo app.blade.php contiene la plantilla de bienvenida a la aplicación cuando no existe una sesión existente.

El archivo cred.blade.php contiene la plantilla para los casos en que se esté creando o editando un recurso.

El archivo index.blade.php contiene la plantilla para los casos en que se esté consultando un recurso de forma general.

El archivo show.blade.php contiene la plantilla para los casos en que se esté consultando un recurso de forma detallada.

El archivo main.blade.php contiene la plantilla general que define disposición de elementos en todas las otras vistas por medio de herencia.

El directorio dashboards contiene las vistas iniciales para los usuarios según su rol:



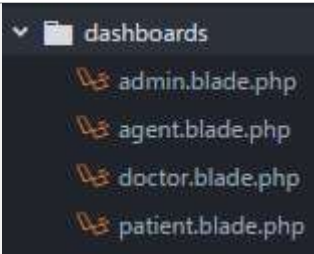
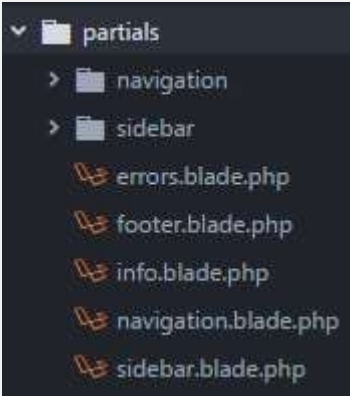
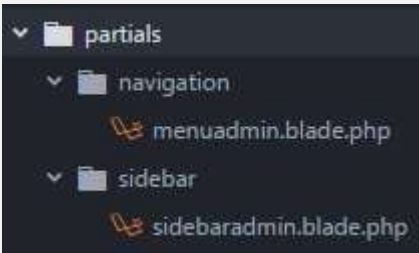
	
	<p>El directorio partials contiene las plantillas reutilizables que serán incrustadas en vistas adicionales ya sea por herencia o inclusión:</p> 
	<p>Los directorios navigation y sidebar contienen las plantillas reutilizables que serán incrustadas en todas las vistas de la aplicación, apoyando en la definición de interacciones según el rol de usuario y la acción que esté desarrollando:</p> 

Tabla 15 Detalle de vistas.  
Encalada, C. (junio 2018).

## Capítulo 5 – Implantación y Aceptación

En este capítulo se detallan las pruebas unitarias que conducen a las pruebas de aceptación, dado que no se construyó un api ni canales que respondan a eventos, se llevaron a cabo pruebas utilizando los diferentes roles especificados por las Historias de Usuario Iniciales y enfocadas a la validación respectiva de los criterios de aceptación.

### 5.1 Pruebas Unitarias

Las siguientes secciones detallan las pruebas unitarias realizadas sobre el producto final, cada tabla incluye una columna que detalla el rol con su correspondiente código http de respuesta y su estado de completamiento; como referencia, los códigos 200 y 403 representan las respuestas “OK” y “Acceso no Permitido” respectivamente.

#### 5.1.1 Paciente

Método	URI	Nombre	Paciente	Completo
GET	/	externals.welcome	200	Si
POST	Citas	Citas.store	200	Si
DELETE	Citas/{appointment}	Citas.destroy	200	Si
GET	Citas/{appointment}	Citas.show	200	Si
GET	Citas/{pid?}/{did?}	Citas.index	200	Si
GET	Citas/create	Citas.create	200	Si
GET	Citas/create/{pid?}/{did?}	Citas.create	200	Si
POST	avatar/{user}	avatar.update	200	Si
GET	changepassword	password.change	200	Si
POST	changepassword	password.update	200	Si
GET	contact	externals.contact	200	Si
POST	contact	externals.reach	200	Si
PUT	contacts/{contact}	contacts.update	200	Si
GET	contacts/{contact}/edit	contacts.edit	200	Si
GET	doctor/{id?}	profile.doctor	200	Si
GET	doctors	doctors.index	200	Si
GET	doctors/{pid?}/{did?}	doctors.index	200	Si
GET	file/{medium}	media.deliver	200	Si
GET	home/{d?}	home	200	Si
GET	locate	externals.locate	200	Si
GET	login	login	200	Si

Tabla 16 Pruebas unitarias para pacientes.  
Encalada, C. (junio 2018).

Método	URI	Nombre	Paciente	Completo
POST	login		200	Si
POST	logout	logout	200	Si
DELETE	media/{medium}	media.destroy	200	Si
GET	media/{medium}	media.show	200	Si
GET	myCitas	Citas.my	200	Si
GET	mydoctors	doctors.my	200	Si
GET	myrecords	records.my	200	Si
POST	password/email	password.email	200	Si
GET	password/reset	password.request	200	Si
POST	password/reset		200	Si
GET	password/reset/{token}	password.reset	200	Si
GET	patient/{id?}	profile.patient	200	Si
GET	patients/{patient}/edit	patients.edit	200	Si
GET	records/{pid}	records.index	200	Si
GET	records/{pid}/{record}	records.show	200	Si
DELETE	records/{record}	records.destroy	200	Si
GET	register	register	200	Si
POST	register		200	Si
GET	schedules	schedules.index	200	Si
GET	schedules/{did}	schedules.show	200	Si
GET	schedules/{did}/{json?}	schedules.list	200	Si
GET	users/{user}/avatar	avatar.edit	200	Si
GET	vitals/{pid}	vitals.index	200	Si
GET	Citas	Citas.index	403	Si
GET	Citas/today/{id?}	Citas.today	403	Si
GET	backup	databases.backup	403	Si
GET	catalogs	catalogs.edit	403	Si
PUT	catalogs	catalogs.update	403	Si
GET	contacts	contacts.index	403	Si
POST	doctors	doctors.store	403	Si
PUT	doctors/{doctor}	doctors.update	403	Si
GET	doctors/{doctor}/edit	doctors.edit	403	Si
GET	doctors/create	doctors.create	403	Si
GET	getbackup	databases.download	403	Si
GET	patients	patients.index	403	Si
PUT	patients/{patient}	patients.update	403	Si

Tabla 16 Pruebas unitarias para pacientes. (cont.)  
Encalada, C. (junio 2018).

<b>Método</b>	<b>URI</b>	<b>Nombre</b>	<b>Paciente</b>	<b>Completo</b>
<b>POST</b>	print	home.print	403	Si
<b>POST</b>	records	records.store	403	Si
<b>PUT</b>	records/{record}	records.update	403	Si
<b>GET</b>	records/{record}/edit	records.edit	403	Si
<b>GET</b>	records/create/{pid}	records.create	403	Si
<b>GET</b>	roles	roles.index	403	Si
<b>POST</b>	roles	roles.store	403	Si
<b>DELETE</b>	roles/{role}	roles.destroy	403	Si
<b>GET</b>	roles/create	roles.create	403	Si
<b>POST</b>	schedules	schedules.store	403	Si
<b>DELETE</b>	schedules/{schedule}	schedules.destroy	403	Si
<b>GET</b>	schedules/create	schedules.create	403	Si
<b>GET</b>	todaycontacts	contacts.today	403	Si
<b>GET</b>	users	users.index	403	Si
<b>DELETE</b>	users/{user}	users.destroy	403	Si
<b>GET</b>	users/{user}	users.show	403	Si
<b>PUT</b>	users/{user}	users.update	403	Si
<b>POST</b>	users/{user}/activate	users.activate	403	Si
<b>GET</b>	users/{user}/edit	users.edit	403	Si
<b>GET</b>	users/{user}/suspend	users.suspend	403	Si
<b>GET</b>	users/create	users.create	403	Si
<b>GET</b>	write/{did?}	home.write	403	Si

Tabla 16 Pruebas unitarias para pacientes. (cont.)  
Encalada, C. (junio 2018).

## 5.1.2 Asistente

<b>Método</b>	<b>URI</b>	<b>Nombre</b>	<b>Asistente</b>	<b>Completo</b>
<b>GET</b>	/	externals.welcome	200	Si
<b>POST</b>	Citas	Citas.store	200	Si
<b>DELETE</b>	Citas/{appointment}	Citas.destroy	200	Si
<b>GET</b>	Citas/{appointment}	Citas.show	200	Si
<b>GET</b>	Citas/{pid?}/{did?}	Citas.index	200	Si
<b>GET</b>	Citas/create	Citas.create	200	Si
<b>GET</b>	Citas/create/{pid?}/{did?}	Citas.create	200	Si
<b>POST</b>	avatar/{user}	avatar.update	200	Si
<b>GET</b>	changepassword	password.change	200	Si
<b>POST</b>	changepassword	password.update	200	Si

Tabla 17 Pruebas unitarias para asistentes.  
Encalada, C. (junio 2018).

Método	URI	Nombre	Asistente	Completo
GET	contact	externals.contact	200	Si
POST	contact	externals.reach	200	Si
PUT	contacts/{contact}	contacts.update	200	Si
GET	contacts/{contact}/edit	contacts.edit	200	Si
GET	doctor/{id?}	profile.doctor	200	Si
GET	doctors	doctors.index	200	Si
GET	doctors/{pid?}/{did?}	doctors.index	200	Si
GET	file/{medium}	media.deliver	200	Si
GET	home/{d?}	home	200	Si
GET	locate	externals.locate	200	Si
GET	login	login	200	Si
POST	login		200	Si
POST	logout	logout	200	Si
DELETE	media/{medium}	media.destroy	200	Si
GET	media/{medium}	media.show	200	Si
GET	myCitas	Citas.my	200	Si
GET	mydoctors	doctors.my	200	Si
GET	myrecords	records.my	200	Si
GET	changepassword	password.change	200	Si
POST	changepassword	password.update	200	Si
GET	contact	externals.contact	200	Si
POST	contact	externals.reach	200	Si
PUT	contacts/{contact}	contacts.update	200	Si
GET	contacts/{contact}/edit	contacts.edit	200	Si
GET	doctor/{id?}	profile.doctor	200	Si
GET	doctors	doctors.index	200	Si
GET	doctors/{pid?}/{did?}	doctors.index	200	Si
GET	file/{medium}	media.deliver	200	Si
GET	home/{d?}	home	200	Si
GET	locate	externals.locate	200	Si
GET	login	login	200	Si
POST	login		200	Si
POST	logout	logout	200	Si
DELETE	media/{medium}	media.destroy	200	Si
GET	media/{medium}	media.show	200	Si

Tabla 17 Pruebas unitarias para asistentes. (cont.)  
Encalada, C. (junio 2018).

<b>Método</b>	<b>URI</b>	<b>Nombre</b>	<b>Asistente</b>	<b>Completo</b>
<b>GET</b>	patient/{id?}	profile.patient	200	Si
<b>GET</b>	patients/{patient}/edit	patients.edit	200	Si
<b>GET</b>	records/{pid}	records.index	200	Si
<b>GET</b>	records/{pid}/{record}	records.show	200	Si
<b>DELETE</b>	records/{record}	records.destroy	200	Si
<b>GET</b>	register	register	200	Si
<b>POST</b>	register		200	Si
<b>GET</b>	schedules	schedules.index	200	Si
<b>GET</b>	schedules/{did}	schedules.show	200	Si
<b>GET</b>	schedules/{did}/{json?}	schedules.list	200	Si
<b>GET</b>	users/{user}/avatar	avatar.edit	200	Si
<b>GET</b>	vitals/{pid}	vitals.index	200	Si
<b>GET</b>	backup	databases.backup	403	Si
<b>GET</b>	catalogs	catalogs.edit	403	Si
<b>PUT</b>	catalogs	catalogs.update	403	Si
<b>GET</b>	contacts	contacts.index	200	Si
<b>POST</b>	doctors	doctors.store	403	Si
<b>PUT</b>	doctors/{doctor}	doctors.update	403	Si
<b>GET</b>	doctors/{doctor}/edit	doctors.edit	403	Si
<b>GET</b>	doctors/create	doctors.create	403	Si
<b>GET</b>	getbackup	databases.download	403	Si
<b>GET</b>	patients	patients.index	200	Si
<b>PUT</b>	patients/{patient}	patients.update	200	Si
<b>POST</b>	print	home.print	200	Si
<b>POST</b>	records	records.store	403	Si
<b>PUT</b>	records/{record}	records.update	403	Si
<b>GET</b>	records/{record}/edit	records.edit	403	Si
<b>GET</b>	records/create/{pid}	records.create	403	Si
<b>GET</b>	todaycontacts	contacts.today	200	Si
<b>GET</b>	users	users.index	403	Si
<b>DELETE</b>	users/{user}	users.destroy	403	Si
<b>GET</b>	users/{user}	users.show	403	Si
<b>PUT</b>	users/{user}	users.update	403	Si
<b>GET</b>	users/create	users.create	200	Si
<b>GET</b>	write/{did?}	home.write	200	Si

Tabla 17 Pruebas unitarias para asistentes. (cont.)  
Encalada, C. (junio 2018).

### 5.1.3 Doctor

Método	URI	Nombre	Doctor	Completo
GET	/	externals.welcome	200	Si
POST	Citas	Citas.store	200	Si
DELETE	Citas/{appointment}	Citas.destroy	200	Si
GET	Citas/{appointment}	Citas.show	200	Si
GET	Citas/{pid?}/{did?}	Citas.index	200	Si
GET	Citas/create	Citas.create	200	Si
GET	Citas/create/{pid?}/{did?}	Citas.create	200	Si
POST	avatar/{user}	avatar.update	200	Si
GET	changepassword	password.change	200	Si
POST	changepassword	password.update	200	Si
GET	contact	externals.contact	200	Si
POST	contact	externals.reach	200	Si
PUT	contacts/{contact}	contacts.update	200	Si
GET	contacts/{contact}/edit	contacts.edit	200	Si
GET	doctor/{id?}	profile.doctor	200	Si
GET	doctors	doctors.index	200	Si
GET	doctors/{pid?}/{did?}	doctors.index	200	Si
GET	file/{medium}	media.deliver	200	Si
GET	home/{d?}	home	200	Si
GET	locate	externals.locate	200	Si
GET	login	login	200	Si
POST	login		200	Si
POST	logout	logout	200	Si
DELETE	media/{medium}	media.destroy	200	Si
GET	media/{medium}	media.show	200	Si
GET	myCitas	Citas.my	200	Si
GET	mydoctors	doctors.my	200	Si
GET	myrecords	records.my	200	Si
POST	password/email	password.email	200	Si
GET	password/reset	password.request	200	Si
POST	password/reset		200	Si
GET	password/reset/{token}	password.reset	200	Si
GET	patient/{id?}	profile.patient	200	Si
GET	patients/{patient}/edit	patients.edit	200	Si

Tabla 18 Pruebas unitarias para doctores.  
Encalada, C. (junio 2018).

Método	URI	Nombre	Doctor	Completo
GET	records/{pid}	records.index	200	Si
GET	records/{pid}/{record}	records.show	200	Si
DELETE	records/{record}	records.destroy	200	Si
GET	register	register	200	Si
POST	register		200	Si
GET	schedules	schedules.index	200	Si
GET	schedules/{did}	schedules.show	200	Si
GET	schedules/{did}/{json?}	schedules.list	200	Si
GET	users/{user}/avatar	avatar.edit	200	Si
GET	vitals/{pid}	vitals.index	200	Si
GET	Citas	Citas.index	200	Si
GET	Citas/today/{id?}	Citas.today	200	Si
GET	backup	databases.backup	403	Si
GET	catalogs	catalogs.edit	403	Si
PUT	catalogs	catalogs.update	403	Si
GET	contacts	contacts.index	200	Si
POST	doctors	doctors.store	403	Si
PUT	doctors/{doctor}	doctors.update	403	Si
GET	doctors/{doctor}/edit	doctors.edit	403	Si
GET	doctors/create	doctors.create	403	Si
GET	getbackup	databases.download	403	Si
GET	patients	patients.index	200	Si
PUT	patients/{patient}	patients.update	200	Si
POST	print	home.print	200	Si
POST	records	records.store	200	Si
PUT	records/{record}	records.update	200	Si
GET	records/{record}/edit	records.edit	200	Si
GET	records/create/{pid}	records.create	200	Si
GET	roles	roles.index	403	Si
POST	roles	roles.store	403	Si
DELETE	roles/{role}	roles.destroy	403	Si
GET	roles/create	roles.create	403	Si
POST	schedules	schedules.store	403	Si
DELETE	schedules/{schedule}	schedules.destroy	403	Si
GET	schedules/create	schedules.create	403	Si

Tabla 18 Pruebas unitarias para doctores. (cont.)  
Encalada, C. (junio 2018).



Método	URI	Nombre	Doctor	Completo
GET	todaycontacts	contacts.today	200	Si
GET	users	users.index	403	Si
DELETE	users/{user}	users.destroy	200	Si
GET	users/{user}	users.show	403	Si
PUT	users/{user}	users.update	403	Si
POST	users/{user}/activate	users.activate	200	Si
GET	users/{user}/edit	users.edit	403	Si
GET	users/{user}/suspend	users.suspend	200	Si
GET	users/create	users.create	200	Si
GET	write/{did?}	home.write	200	Si

Tabla 18 Pruebas unitarias para doctores. (cont.)  
Encalada, C. (junio 2018).

#### 5.1.4 Administrador

Método	URI	Nombre	Administrador	Completo
GET	/	externals.welcome	200	Si
POST	Citas	Citas.store	200	Si
DELETE	Citas/{appointment}	Citas.destroy	200	Si
GET	Citas/{appointment}	Citas.show	200	Si
GET	Citas/{pid?}/{did?}	Citas.index	200	Si
GET	Citas/create	Citas.create	200	Si
GET	Citas/create/{pid?}/{did?}	Citas.create	200	Si
POST	avatar/{user}	avatar.update	200	Si
GET	changepassword	password.change	200	Si
POST	changepassword	password.update	200	Si
GET	contact	externals.contact	200	Si
POST	contact	externals.reach	200	Si
PUT	contacts/{contact}	contacts.update	200	Si
GET	contacts/{contact}/edit	contacts.edit	200	Si
GET	doctor/{id?}	profile.doctor	200	Si
GET	doctors	doctors.index	200	Si
GET	doctors/{pid?}/{did?}	doctors.index	200	Si
GET	file/{medium}	media.deliver	200	Si
GET	home/{d?}	home	200	Si
GET	locate	externals.locate	200	Si

Tabla 19 Pruebas unitarias para administradores.  
Encalada, C. (junio 2018).

Método	URI	Nombre	Administrador	Completo
GET	login	login	200	Si
POST	login		200	Si
POST	logout	logout	200	Si
DELETE	media/{medium}	media.destroy	200	Si
GET	media/{medium}	media.show	200	Si
GET	myCitas	Citas.my	200	Si
GET	mydoctors	doctors.my	200	Si
GET	myrecords	records.my	200	Si
POST	password/email	password.email	200	Si
GET	password/reset	password.request	200	Si
POST	password/reset		200	Si
GET	password/reset/{token}	password.reset	200	Si
GET	patient/{id?}	profile.patient	200	Si
GET	patients/{patient}/edit	patients.edit	200	Si
GET	records/{pid}	records.index	200	Si
GET	records/{pid}/{record}	records.show	200	Si
DELETE	records/{record}	records.destroy	200	Si
GET	register	register	200	Si
POST	register		200	Si
GET	schedules	schedules.index	200	Si
GET	schedules/{did}	schedules.show	200	Si
GET	schedules/{did}/{json?}	schedules.list	200	Si
GET	users/{user}/avatar	avatar.edit	200	Si
GET	vitals/{pid}	vitals.index	200	Si
GET	Citas	Citas.index	200	Si
GET	Citas/today/{id?}	Citas.today	200	Si
GET	backup	databases.backup	200	Si
GET	catalogs	catalogs.edit	200	Si
PUT	catalogs	catalogs.update	200	Si
GET	contacts	contacts.index	200	Si
POST	doctors	doctors.store	200	Si
PUT	doctors/{doctor}	doctors.update	200	Si
GET	doctors/{doctor}/edit	doctors.edit	200	Si
GET	doctors/create	doctors.create	200	Si
GET	getbackup	databases.download	200	Si

Tabla 19 Pruebas unitarias para administradores. (cont.)  
Encalada, C. (junio 2018).

Método	URI	Nombre	Administrador	Completo
GET	patients	patients.index	200	Si
PUT	patients/{patient}	patients.update	200	Si
POST	print	home.print	200	Si
POST	records	records.store	200	Si
PUT	records/{record}	records.update	200	Si
GET	records/{record}/edit	records.edit	200	Si
GET	records/create/{pid}	records.create	200	Si
GET	roles	roles.index	200	Si
POST	roles	roles.store	200	Si
DELETE	roles/{role}	roles.destroy	200	Si
GET	roles/create	roles.create	200	Si
POST	schedules	schedules.store	200	Si
DELETE	schedules/{schedule}	schedules.destroy	200	Si
GET	schedules/create	schedules.create	200	Si
GET	todaycontacts	contacts.today	200	Si
GET	users	users.index	200	Si
DELETE	users/{user}	users.destroy	200	Si
GET	users/{user}	users.show	200	Si
PUT	users/{user}	users.update	200	Si
POST	users/{user}/activate	users.activate	200	Si
GET	users/{user}/edit	users.edit	200	Si
GET	users/{user}/suspend	users.suspend	200	Si
GET	users/create	users.create	200	Si
GET	write/{did?}	home.write	200	Si

Tabla 19 Pruebas unitarias para administradores. (cont.)  
Encalada, C. (junio 2018).

## 5.2 Aceptación

La aceptación del proyecto se refiere a la validación de la funcionalidad detallada en las historias de usuario, dicha validación recibe también el nombre de pruebas de aceptación, las pruebas de aceptación son llevadas a cabo por el cliente o usuario final, en este caso particular los documentos que definen la aceptación del producto se encuentran en la sección de anexos en orden ascendente conforme a la versión a la que corresponden, finalmente se tiene el documento de aceptación del proyecto, mismo que corresponde a la versión 1.0 y cuyo propósito es certificar que todas las pruebas condujeron a resultados satisfactorios.

## **Capítulo 6 –Conclusiones y Recomendaciones**

En la siguiente sección se detalla el cumplimiento de los objetivos del proyecto en orden secuencial para de esta forma poder cubrir el objetivo general en el alcance definido.

### **6.1 Conclusiones**

- Se completó con éxito el análisis de los requerimientos para el desarrollo e implementación del sistema requerido por el Centro de Especialidades Médicas.
- Se crearon satisfactoriamente los módulos para administración de pacientes, archivos multimedia, doctores, usuarios, roles y citas.
- Se realizaron las pruebas unitarias y de aceptación con resultados satisfactorios.
- Se desarrolló un sistema modular, conceptualmente estable y que cubre adecuadamente las necesidades de información del Centro de Especialidades Médicas (sección 3.1).
- Se implantó exitosamente el sistema desarrollado, mismo que se encuentra disponible en la dirección “ceesmed.com”.
- Se logró utilizar los conceptos de Extreme Programming para el desarrollo del proyecto (sección 2.1.2).
- Se logró cubrir los objetivos del proyecto utilizando las herramientas detalladas en la fundamentación teórica (sección 2.2).

### **6.2 Recomendaciones**

- Mantener el diseño modular aplicado a las interfaces de usuario en versiones sucesivas.
- Incrementar la funcionalidad del sistema en relación a la gestión de signos vitales de los pacientes.
- Utilizar control de versionamiento y gestión de configuración en desarrollos posteriores basados en este sistema.
- Utilizar control de versionamiento y gestión de configuración en desarrollos posteriores basados en este sistema.

## Glosario

**Amazon Elastic Beanstalk:** Es un servicio de infraestructura bajo demanda que orquesta la operación de varios componentes en AWS (Amazon Web Services) incluyendo EC2 (Elastic Compute Cloud), S3 (Simple Storage Service), SNS (Servicio simple de notificación), CloudWatch, autoescalado y Elastic Load Balancers. Su objetivo es proporcionar una capa de abstracción sobre el servidor y su sistema operativo de manera que los usuarios vean solo una combinación lista para usar de SO y plataforma.

**Amazon Elastic Load Balancer:** Es un balanceador de carga que sirve como único punto de contacto para los clientes. El balanceador distribuye el tráfico de la aplicación entrante entre múltiples destinos según se haya especificado.

**Amazon RDS:** Relational Database Service es un servicio web que facilita la configuración, operación y ampliación de una base de datos relacional en la nube. Proporciona capacidad variable para una base de datos relacional estándar y gestiona tareas comunes de administración de bases de datos.

**Amazon Route53:** Es un servidor de nombres de dominio (DNS) escalable y altamente disponible. Lanzado el 5 de diciembre de 2010, forma parte de la plataforma de computación en la nube de Amazon.com y AWS. El nombre es una referencia al puerto TCP o UDP 53, donde se abordan las solicitudes del servidor DNS.

**Amazon S3:** Simple Storage Service es un servicio de almacenamiento de archivos en Internet, está diseñado para hacer que la gestión de archivos a escala web sea más fácil para los desarrolladores. Amazon S3 tiene una interfaz simple de servicios web que puede usar para almacenar y recuperar cualquier cantidad de datos, en cualquier momento, desde cualquier lugar.

**Capa de Presentación:** En un sistema distribuido, la capa de presentación se refiere al nivel de un sistema que se ocupa de gestionar la interfaz gráfica del mismo.

**Cohesión:** En programación, la cohesión se refiere al grado en que los elementos dentro de un módulo dependen entre uno del otro, es decir, es una medida de la fuerza de la relación entre los métodos de la clase y los datos que gestiona.

**jQuery:** Es una librería de código JavaScript rápida, pequeña y rica en funciones para el recorrido y manipulación de documentos HTML, manejo de eventos, animación, y ejecución asincrónica; está orientada para la compatibilidad a través de múltiples navegadores.

**Licencia GPL:** La Licencia Pública General de GNU es una licencia de software gratuita sobre copyleft y otros tipos de trabajos.

**Metadatos:** Son datos informativos que acompañan a otros datos con el objetivo de definir estos últimos. Existen muchos tipos de metadatos según sea la naturaleza de la información que acompañan ya sean descriptivos, estructurales, administrativos, de referencia, estadísticos, etc.

**Modelo en cascada (Waterfall):** Enfoque de desarrollo de software lineal, tiende a ser uno de los enfoques menos iterativos y flexibles, ya que el progreso fluye en una sola dirección a través de las etapas de concepción, análisis, diseño, construcción, pruebas, despliegue y mantenimiento.

**Metodologías Ágiles:** Describen un enfoque para el desarrollo de software según el cual los requisitos y las soluciones evolucionan a través del esfuerzo colaborativo en equipos auto-organizados y multifuncionales que también incluyen a los clientes o usuarios finales.

**Modelo, Vista, Controlador (MVC):** Es un patrón de desarrollo de software que desacopla los componentes principales, permitiendo la reutilización eficiente del código y el desarrollo paralelo, divide una aplicación en tres partes (capas) interconectadas con el objetivo de separar el manejo interno y la presentación de información.

**Open SSL:** Librería para criptografía de propósito general que incluye un kit de herramientas robusto y de grado comercial para utilizar en los protocolos TLS (Transport Layer Security) y SSL (Secure Sockets Layer).

**Object Relational Mapping (ORM):** Es una técnica para gestionar y convertir datos entre sistemas de tipos incompatibles que utilizan lenguajes de programación diferentes, consiste en crear una capa de abstracción adicional para manejar datos a través de instancias de clases o modelos determinados desde el lenguaje de programación.

**Programación Orientada a Objetos (OOP):** Es un paradigma de programación de software cuyo objetivo es definir las propiedades y el comportamiento de objetos mediante la creación de clases.

**Pruebas de Aceptación:** (Pruebas de caja negra o pruebas funcionales) son pruebas formales realizadas para determinar si un sistema cumple o no con sus criterios de aceptación según la perspectiva del cliente o usuario final.

**Pruebas Unitarias:** Nivel de prueba de software donde las unidades/componentes individuales de un software son evaluadas para validar que funcionen según como fueron diseñadas. Una unidad es la parte más pequeña comprobable de cualquier software. Por lo general, tiene una o algunas entradas y generalmente una única salida.

**RDBMS:** Es un sistema de gestión de bases de datos (DBMS) basado en el modelo relacional.

**Versionamiento:** Proceso de asignar nombres de versión únicos o números de versión únicos a estados únicos del software dentro de una categoría de número de versión dada, estos números se asignan en orden creciente y corresponden a los desarrollos nuevos que se realizan sobre en el software.

## Referencias

- Beati, H. (2015). *HTML5 y CSS3 para diseñadores*. Bogotá, Colombia: Alfaomega.
- Cabezas Granado, L. M. (2014). *Desarrollo web con PHP y MySQL*. Madrid, España: Anaya Multimedia.
- Group, T. P. (s.f.). *PHP.net Documentation*. Obtenido de php.net: <http://php.net>
- Mark Otto, C. S. (2017). <https://getbootstrap.com/>. Obtenido de Bootstrap: <https://getbootstrap.com>
- Michael Smith, T. N. (2017). *JavaScript*. Obtenido de developer.mozilla.org: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- Muhammad Ali Babar, A. W. (2014). Extreme Programming. En *Agile Software Architecture, Aligning Agile Processes and Software Architectures* (pág. 13). Waltham, MA: Elsevier.
- Otwell, T. (2017). *Laravel Documentation*. Obtenido de Laravel.com: <https://laravel.com/docs/5.5>
- Silvia Rivadeneira, G. V. (2013). *El modelado de requerimientos en las metodologías ágiles*. Obtenido de Repositorio Institucional de la UNLP: <http://sedici.unlp.edu.ar/bitstream/10915/27196/1/El+modelado+de+requerimientos+en+las+metodolog%C3%ADas+%C3%A1giles>
- What is PHP?* (2017). Obtenido de Php.net: <http://php.net/manual/en/intro-what-is.php>



## Anexos

### Actas de Reunión

#### Reunión #1

##### Lugar

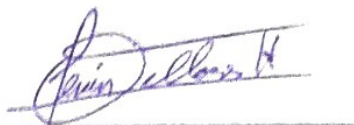
- Humberto Albornoz Oe8-181 y Atacames, Centro de Especialidades Médicas

##### Fecha y Hora

- 1 de Diciembre de 2017 - 18:30

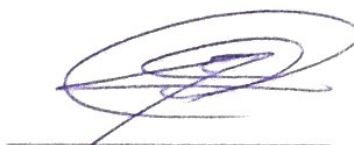
##### Temas Tratados

- Recopilación de información inicial.
- Análisis de Requerimientos.
- Formulación de historias de usuario iniciales.
- Realización de la Carta de Compromiso del Centro de Especialidades Médicas para el desarrollo del proyecto.



---

Dr. Lenin Villacrés



---

Carlos Encalada

## Reunión #2

### Lugar

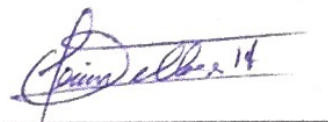
- Humberto Albornoz Oe8-181 y Atacames, Centro de Especialidades Médicas

### Fecha y Hora

- 22 de diciembre de 2017 - 18:30

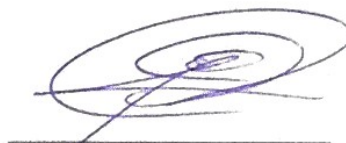
### Temas Tratados

- Recopilación de información adicional.
- Formulación de planificación.
- Formulación de estimaciones para historias de usuario iniciales.



---

Dr. Lenin Villacrés



---

Carlos Encalada

## Reunión #3

### Lugar

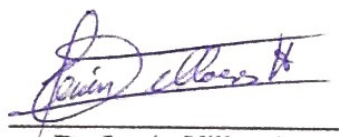
- Humberto Albornoz Oe8-181 y Atacames, Centro de Especialidades Médicas

### Fecha y Hora

- 12 de Enero de 2018 - 18:30

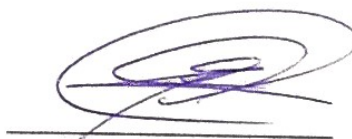
### Temas Tratados

- Presentación de esquema de interfaces.
- Análisis de información requerida en la vista index de cada recurso.
- Definición de esquema de autorizaciones.



---

Dr. Lenin Villacrés



---

Carlos Encalada

## Reunión #4

### Lugar

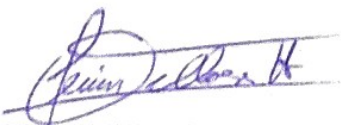
- Humberto Albornoz Oe8-181 y Atacames, Centro de Especialidades Médicas

### Fecha y Hora

- 2 de Febrero de 2018 - 18:30


### Temas Tratados

- Presentación de interfaces de usuario.
- Definición de temas puntuales para mejorar la experiencia de usuario.
- Presentación del esquema de autorizaciones.



---

Dr. Lenin Villacrés



---

Carlos Encalada

## Reunión #5

### Lugar

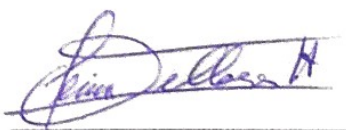
- Humberto Albornoz Oe8-181 y Atacames, Centro de Especialidades Médicas

### Fecha y Hora

- 23 de Febrero de 2018 - 18:30

### Temas Tratados

- Presentación de módulos para gestión de pacientes, contactos.
- Diseño de esquema para dashboards de inicio según los roles.
- Explicación de costos asociados a la implantación.



Dr. Lenin Villacrés



Carlos Encalada

## Reunión #6

### Lugar

- Humberto Albornoz Oe8-181 y Atacames, Centro de Especialidades Médicas

### Fecha y Hora

- 16 de Marzo de 2018 - 18:30

### Temas Tratados

- Generación de Interfaz individualizada para registro de pacientes nuevos.
- Presentación de resultados de Pruebas Unitarias.
- Presentación del software implantado en el servidor.
- Inicio de Periodo para Pruebas de aceptación.
- Presentación de aplicación de esquema de autorizaciones y restricciones adicionales.



Dr. Lenin Villacrés



Carlos Encalada

## Reunión #7

### Lugar

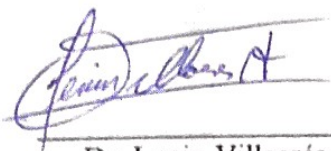
- Humberto Albornoz Oe8-181 y Atacames, Centro de Especialidades Médicas

### Fecha y Hora

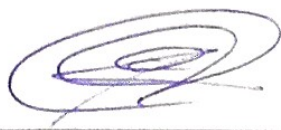
- 20 de Abril de 2018 - 18:30

### Temas Tratados

- Documento de Pruebas de Aceptación.
- Entrega del Producto.
- Presentación de Manual Técnico y Manual de Usuario.



Dr. Lenin Villacrés



Carlos Encalada

## Documento de Aceptación

### Aceptación del Producto

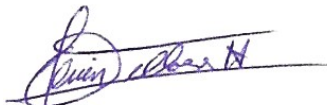
Quito, 18 de junio de 2018

Sr. Dr. Gustavo Chafla Altamirano  
Decano de la Facultad De Ingeniería  
PUCE

De mi consideración,

Yo, Lenin Villacrés, con C.I. 123456789, Director del CENTRO DE ESPECIALIDADES MÉDICAS, luego del respectivo periodo de estabilización del software y la instalación del mismo en ambiente de producción, confirmo el cumplimiento de los requerimientos establecidos al inicio del proyecto con resultados satisfactorios, en tal razón doy por concluido el mismo y acepto que el software construido por el Sr. Carlos Encalada con C.I. 1712507654 ha sido correctamente implantado.

Atentamente,



---

Dr. Lenin Villacrés  
Director  
Centro de Especialidades Médicas



# Manual Técnico

## Introducción

Este documento tiene el propósito de servir como guía sobre de las principales características y actividades técnicas relacionadas con la aplicación, para de esta manera asistir al usuario técnico en la instalación y configuración de la misma.

## Roles

A continuación, se presenta el diagrama de jerarquía de roles en relación con la funcionalidad de la aplicación.

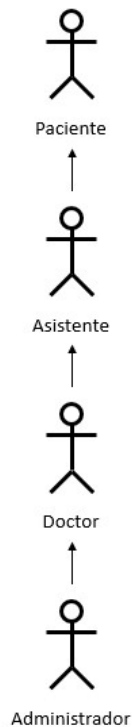


Fig. 1 Diagrama de Jerarquía de Usuarios.  
Encalada, C. (junio 2018).

### Paciente

El paciente puede realizar actividades sobre su perfil y sus citas.

- Gestionar cuenta
- Gestionar perfil
- Gestionar citas

### Asistente

El asistente puede gestionar las citas de los pacientes y médicos, así como consultar la información de contacto y disponibilidad de los médicos.

- Gestionar citas
- Gestionar contactos
- Gestionar pacientes

### Doctor

El doctor tiene la capacidad de gestionar registros médicos.

- Gestionar citas
- Gestionar contactos
- Gestionar pacientes
- Gestionar registros

### Administrador

El administrador gestiona las características generales del sistema, sus usuarios, roles y respaldo de datos.

- Gestionar citas
- Gestionar contactos
- Gestionar doctores
- Gestionar horarios
- Gestionar pacientes
- Gestionar registros
- Gestionar roles
- Gestionar configuración

## Software de Terceros

A continuación, se detalla las librerías externas al proyecto que fueron utilizadas para la capa de presentación.

### Bootstrap

Se utiliza en todas las vistas del proyecto, la versión seleccionada es 4.0.

### Charts.js

Se utiliza en los perfiles de paciente para dibujar las gráficas de estatura, crecimiento, relación peso/talla e índice de masa corporal.

## Diccionario de Datos

A continuación, se muestra el detalle de los campos construidos en la base de datos según su tipo y tabla a la que pertenecen.

Tabla	Campo	Tipo	Requerido
MEDIA	Id	int - auto_increment	Si
	Mime	char(255)	Si
	Path	text	Si
RECORDS	Id	int	Si
	patient_id	int	Si
	Description	char(200)	Si
	Diagnosis	char(200)	Si
	prescription	char(200)	Si
	created_at	timestamp	Si
	updated_at	timestamp	Si
APPOINTMENTS	Id	int	Si
	doctor_ID	int	Si
	patient_ID	int	Si
	Time	timestamp	Si
	Type	char(20)	Si
	Reason	char(200)	Si

Tabla 1 Diccionario de datos.  
Encalada, C. (junio 2018).

Tabla	Campo	Tipo	Requerido
USERS	Id	int	Si
	nid	char(13)	Si
	Name	char(50)	Si
	Email	char(50)	Si
	Password	char(100)	Si
	remember_token	char(100)	Si
	Active	bool	Si
DOCTORS	Id	int	Si
	user_id	int	Si
	Specialty	char(50)	Si
	Position	char(100)	Si
CONTACTS	user_id	int	Si
	Pone	char(12)	Si
	Mobile	char(12)	Si
	Address	char(100)	Si
	City	char(30)	Si
SCHEDULES	Id	int - auto_increment	Si
	doctor_id	int	Si
	day	char(3)	Si
	start	time	Si
	end	time	Si
VITALS	record_id	int	Si
	pulse	int	Si
	bpsystolic	decimal(5,2)	Si
	bpdiastolic	decimal(5,2)	Si
	temperature	decimal(4,2)	Si
	weight	decimal(5,2)	Si
	height	decimal(5,2)	Si
ROLES	id	int	Si
	name	char(30)	Si
	description	char(50)	Si
PATIENTS	user_id	int	Si
	birthdate	date	Si
	gender	char(1)	Si
	bloodtype	char(3)	Si
	notes	text	Si
	viewhistory	bool	Si

Tabla 1 Diccionario de datos (cont.).  
Encalada, C. (junio 2018).

## Instalación y configuración

En esta sección se describe el proceso de instalación de la aplicación.

Requerimientos generales:

- El dominio debe estar debidamente asociado al servidor, en este caso, el dominio “ceesmed.com” debe apuntar al servidor cuya ip es 18.221.232.252; se puede verificar esto haciendo ping a la dirección alternativa de la aplicación: “ping ceesmedtst.us-east-2.elasticbeanstalk.com”, ambas direcciones IP deberían ser idénticas, caso contrario el administrador deberá verificar los conjuntos de registros en la consola de administración de Amazon Route53.
- El servidor de aplicación debe poder conectarse con el de base de datos, es decir, las reglas de conectividad de la base de datos deben permitir que existan conexiones desde la dirección IP del servidor, esta configuración se puede verificar en la sección “security groups” de la consola de administración de instancia en el servicio de Amazon RDS.

Requerimientos del servidor de aplicación:

- Servidor web con soporte para php
- PHP V7.0 o superior
- 20 megabytes de espacio en disco

Requerimientos del servidor de base de datos:

- MySQL V5.1 o superior.

Configuración:

Para poder iniciar la aplicación se deben verificar los valores que constan en el archivo “.env“, mismo que está localizado en el directorio raíz de la aplicación y a continuación verificar que el servicio Amazon ElasticBeanstalk registre como punto de entrada el archivo “index.php” localizado en la carpeta “Public”, esto es un estándar en todas las aplicaciones creadas con Laravel.

<b>Campo</b>	<b>Nombre</b>	<b>Descripción y valor</b>
<b>ADMIN_NAME</b>	Nombre del usuario administrador	Nombre de la persona que administrará la aplicación, este campo podrá ser cambiado desde las opciones de perfil del usuario.
<b>ADMIN_EMAIL</b>	Email del usuario administrador	Email de la persona que administrará la aplicación, este campo será utilizado para propósitos de autenticación y podrá ser cambiado desde las opciones de perfil del usuario.
<b>APP_NAME</b>	Nombre de la aplicación	Nombre con el que los usuarios asociará a la aplicación, puede estar vinculado a un wordmark o lettermark, este campo es utilizado para propósitos de identidad de marca o similares.
<b>APP_ENV</b>	Entorno de la aplicación	Entorno sobre el cual funciona la aplicación, actualmente el valor es “PROD”, correspondiente a producción.
<b>APP_URL</b>	Dominio	Nombre del dominio asociado a la IP del servidor, este campo se utiliza para la generación de URLs.
<b>APP_TIMEZONE</b>	Zona horaria	Zona horaria sobre la cual está basada la aplicación, este campo se utiliza para calcular las fechas en la gestión de citas y horarios.

Tabla 2 Parámetros listados en el archivo “env”.  
Encalada, C. (junio 2018).

<b>Campo</b>	<b>Nombre</b>	<b>Descripción y Valor</b>
<b>DB_CONNECTION</b>	Conexión de base de datos	Nombre del controlador asociado para la conexión con el servidor de base de datos, actualmente este valor es “mysql”.
<b>DB_HOST</b>	Dirección del servidor de base de datos	Este valor corresponde a la IP asignada al servidor de base de datos, también se puede especificar la dirección de dominio, actualmente este campo tiene el valor “ceesmedrds.cdy5c7s6ccff.us-east-2.rds.amazonaws.com”.
<b>DB_PORT</b>	Puerto del servidor de base de datos	Puerto utilizado para la conexión TCP hacia el servidor de base de datos, actualmente este campo tiene el valor “3306”.
<b>DB_DATABASE</b>	Nombre de la base de datos reservada	Se asume que el servidor de base de datos puede contener más de una base de datos, por ello es necesario especificar el nombre asociado al esquema sobre el que funciona la aplicación, actualmente este campo tiene el valor “ceesmed”.
<b>DB_USERNAME</b>	Usuario	Nombre de usuario para acceder al servidor de base de datos: “ceesmed”
<b>DB_PASSWORD</b>	Contraseña	Contraseña para acceder al servidor de base de datos: “890()=iopIOP”

Tabla 2 Parámetros listados en el archivo “env” (cont.).  
Encalada, C. (junio 2018).

Una vez editados los campos se debe iniciar el servidor web, ya sea este una máquina virtual o un servidor físico. Con el servidor ya configurado, se deberá acceder a la consola artisan y se ejecutará el siguiente comando: “php artisan migrate --seed”, éste comando dará inicio al proceso de creación de las estructuras necesarias en la base de datos y poblará la misma con los datos del usuario administrador.

A continuación, se podrá acceder a la aplicación mediante el dominio asociado a la misma y se podrá iniciar sesión con las credenciales especificadas para el usuario administrador.

# Manual de usuario

## Pantalla inicial



Fig. 1 Página de Bienvenida.  
Encalada, C. (junio 2018).

Se encontrará dos opciones, “Contact” y “Location”, la primera opción es para que el usuario del sistema se comuniqué con el administrador del mismo (Fig. 2) y la segunda para que pueda ver la ubicación del centro médico (Fig. 3).

The screenshot shows a contact form titled 'Centro de Especialidades Médicas'. The form contains four input fields: 'Name', 'E-Mail', 'Phone', and 'Message'. The 'Name' field is highlighted with a blue border. Below the input fields, there are two buttons: 'Send' and 'Cancel'.

Fig. 2 Formulario de Contacto.  
Encalada, C. (junio 2018).



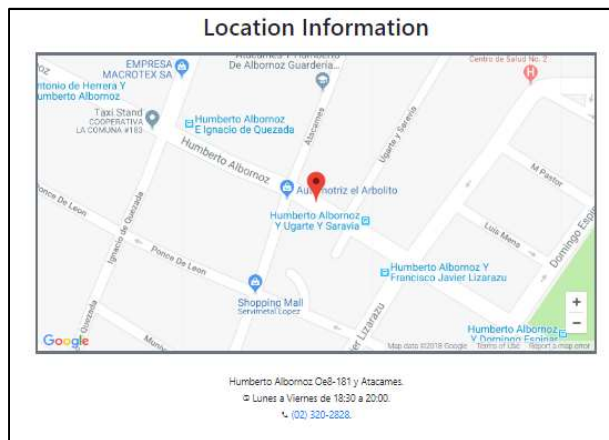


Fig. 3 Información de Ubicación.  
Encalada, C. (junio 2018).

## Registro de usuario

Acceder a la página de bienvenida (Fig. 1) y seleccionar la opción “Register” (Fig. 4).

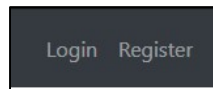


Fig. 4 Opciones existentes en la pantalla de bienvenida.  
Encalada, C. (junio 2018).

Llenar el formulario de registro con los datos de usuario correspondientes (Fig. 5).

**Register**

Name

ID Number

E-Mail Address

Password

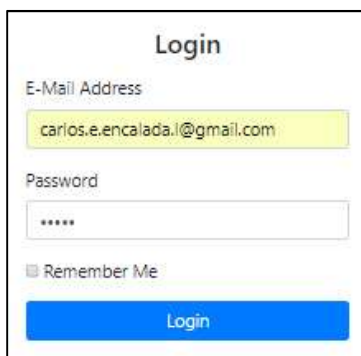
Confirm Password

**Register**

Fig. 5 Formulario de Registro.  
Encalada, C. (junio 2018).

## Inicio de sesión

Ingresa a la página de bienvenida (Fig. 1) y seleccionar la opción “Login”, en el formulario de inicio de sesión ingresar los datos previamente registrados en el sistema (Fig. 6).



The image shows a login form titled "Login". It contains two input fields: "E-Mail Address" with the value "carlos.e.encalada.l@gmail.com" and "Password" with masked characters "\*\*\*\*\*". Below the password field is a checkbox labeled "Remember Me". At the bottom is a blue button labeled "Login".

Fig. 6 Formulario de Inicio de Sesión.  
Encalada, C. (junio 2018).

Por defecto, el sistema presentará la pantalla inicial para el rol de paciente (Fig. 7), si el usuario tuviese roles adicionales, las opciones correspondientes se encontrarán en el menú de roles en la barra superior (Fig. 8).

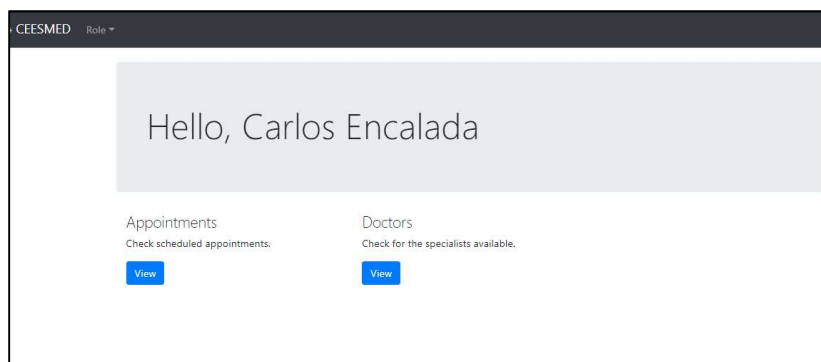


Fig. 7 Pantalla Inicial de Paciente.  
Encalada, C. (junio 2018).

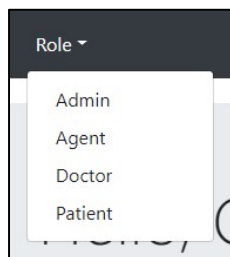


Fig. 8 Menú desplegable para roles.  
Encalada, C. (junio 2018).

## Opciones para los roles

El sistema tiene los siguientes roles disponibles: Administrador, Asistente, Doctor y Paciente. Cada rol tendrá una funcionalidad específica definida accesible ya sea desde la pantalla inicial del rol (Fig. 7) o desde la barra lateral del sistema (Fig. 9).

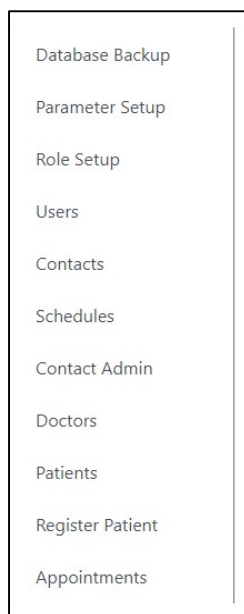


Fig. 9 Barra Lateral.  
Encalada, C. (junio 2018).

## Respaldo de datos

Para realizar un respaldo de datos a demanda, se deberá acceder a la opción “Database Backup” en la barra lateral (Fig. 9), cuando el respaldo haya sido construido el sistema mostrará un mensaje informativo con la opción para descargar el respaldo de datos.

## Configurar parámetros

Para configurar los parámetros del sistema, el usuario deberá acceder a la opción “Parameter Setup” de la barra lateral (Fig. 9) y proceder a editar los valores correspondientes en el formulario de edición de parámetros (Fig. 10), una vez se haya completado la edición, presionar el botón de guardado (Fig. 10) para persistir los cambios en el sistema.

Fig. 10 Formulario de Edición de Parámetros.  
Encalada, C. (junio 2018).

## Gestión de Roles

El usuario administrador del sistema podrá asignar roles a otros usuarios según lo requiera, para ello deberá acceder a la opción “Role Setup” de la barra lateral (Fig. 9) y presionar el botón “New Role” de la vista de listado de roles (Fig. 11), en el formulario de asignación deberá seleccionar el usuario y el rol que se le desea asignar (Fig. 12) y finalmente presionar el botón de guardado para persistir los cambios en el sistema.

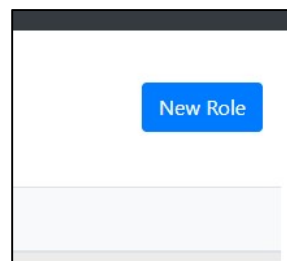


Fig. 11 Botón para acceder al formulario de asignación.  
Encalada, C. (junio 2018).

Fig. 12 Formulario de asignación de roles.  
Encalada, C. (junio 2018).

Si el usuario nuevo fuera a recibir el rol de doctor, el sistema lo dirigirá automáticamente al formulario de registro de doctores (Fig. 13), una vez que se hayan llenado los campos y presionado el botón de guardado, el doctor será registrado en el sistema y el correspondiente rol quedará asignado de forma automática.

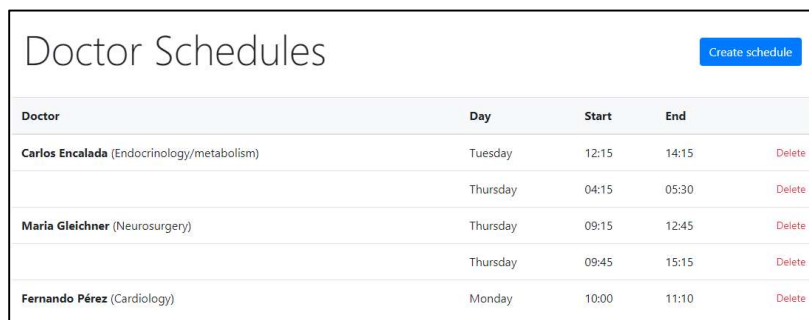


The image shows a web form titled "Create doctor". It contains three dropdown menus labeled "User", "Specialty", and "Position". Below these menus are two buttons: "Save" and "Cancel".

Fig. 13 Formulario de Registro de Doctor.  
Encalada, C. (junio 2018).

## Gestión de Horarios

Para crear o eliminar horarios el usuario administrador deberá acceder al Índice de Horarios (Fig. 14), para ello se debe seleccionar la opción “Schedules” de la barra lateral (Fig. 9).



The image shows a table titled "Doctor Schedules" with a "Create schedule" button in the top right corner. The table has four columns: "Doctor", "Day", "Start", and "End". It lists schedules for three doctors: Carlos Encalada, Maria Gleichner, and Fernando Pérez.

Doctor	Day	Start	End
Carlos Encalada (Endocrinology/metabolism)	Tuesday	12:15	14:15
	Thursday	04:15	05:30
Maria Gleichner (Neurosurgery)	Thursday	09:15	12:45
	Thursday	09:45	15:15
Fernando Pérez (Cardiology)	Monday	10:00	11:10

Fig. 14 Índice de Horarios.  
Encalada, C. (junio 2018).

Los horarios se deben asignar de forma personalizada para cada doctor, para crear un horario se debe seleccionar la opción “Create Schedule” en la parte superior derecha del índice de horarios (Fig. 14), llenar los campos del formulario de creación de horario (Fig. 15) y presionar el botón de guardado para persistir los cambios.

**Create schedule**

Doctor

Day

Start

End

Fig. 15 Formulario de Creación de Horario.  
 Encalada, C. (junio 2018).

El sistema validará los datos y el nuevo horario se mostrará en el índice de horarios (Fig. 14) agrupado según el especialista al que pertenezca. Si los datos fueran inconsistentes o el nuevo horario estuviese traslapado con otro del mismo especialista, el sistema mostrará los mensajes de error pertinentes.

## Gestión de Usuarios

Para registrar un usuario, acceder a la opción “Register Patient” de la barra lateral (Fig. 9), llenar los campos del formulario de registro de paciente (Fig. 16) y presionar el botón de guardado.

**New Patient**

Name:  ID:

Email:  Date of birth:

Gender:  Bloodtype:

Background

Pathological History:

Allergies: NA

Vaccines: NA

Other:

Phone:  Mobile:

Address:  City:

Fig. 16 Formulario de Registro de Paciente.  
 Encalada, C. (junio 2018).

Si los datos son correctos el sistema procederá a guardar los datos del nuevo paciente y posteriormente el sistema redirigirá al usuario hacia el perfil del nuevo paciente (Fig. 17), éste último aparecerá en el Índice de Pacientes (Fig. 18) y en el Índice de Usuarios (Fig. 19).

Fig. 17 Perfil de paciente.  
Encalada, C. (junio 2018).

La información del paciente es accesible de forma centralizada desde su perfil asignado, de modo que, si se necesitara modificar información particular sobre éste, esto fuese fácilmente realizable mediante los atajos provistos en esta vista (Fig. 17), desde esta vista también se accede a la creación de registros médicos para el paciente y a su historia personal (parte inferior izquierda, Fig. 17).

ID	Name	Email	Updated	
7125331013	Richmond Schinner DDS	pheller@example.net	June 25, 2018 01:15	<a href="#">Details</a> <a href="#">Update</a> <a href="#">Suspend</a>
1792029324	Yolanda	yolanda_macias@hotmail.com	June 24, 2018 22:03	<a href="#">Details</a> <a href="#">Update</a> <a href="#">Activate</a>
1792029322	Fernando Pérez	fernandoperezlopez82@outlook.com	June 24, 2018 21:15	<a href="#">Details</a> <a href="#">Update</a> <a href="#">Suspend</a>
8204116659	Johnathon Fay V	nolan.leanne@example.org	June 24, 2018 14:04	<a href="#">Details</a> <a href="#">Update</a> <a href="#">Suspend</a>
	Dummy Patient	no email	June 24, 2018 14:02	<a href="#">Details</a> <a href="#">Update</a> <a href="#">Activate</a>
9999999	Beatriz Campos	bcampos@puce.edu.ec	June 24, 2018 13:52	<a href="#">Details</a> <a href="#">Update</a> <a href="#">Suspend</a>
000000020	Lenin Villacres	leninvillacres@gmail.com	June 24, 2018 12:56	<a href="#">Details</a> <a href="#">Update</a> <a href="#">Suspend</a>
000000000	Carlos Encalada	carlos.e.encalada.l@gmail.com	June 24, 2018 00:35	<a href="#">Details</a> <a href="#">Update</a> <a href="#">Suspend</a>
8680644755	Maria Gleichner	hollis67@example.com	June 24, 2018 00:35	<a href="#">Details</a> <a href="#">Update</a> <a href="#">Activate</a>

Fig. 18 Índice de Usuarios.  
Encalada, C. (junio 2018).

Patient Index					Name or Email		Search
Name	Date of birth	Gender	History Enabled	Bloodtype			
Beatriz Campos	January 1, 1970	female	no	*	<a href="#">Profile</a>	<a href="#">Update</a>	
Carlos Encalada	October 2, 2012	male	no	O -	<a href="#">Profile</a>	<a href="#">Update</a>	
Dixie Kassulke	June 5, 2015	undisclosed	no	AB*	<a href="#">Profile</a>	<a href="#">Update</a>	
Dummy Patient	June 21, 2018	male	no	*	<a href="#">Profile</a>	<a href="#">Update</a>	
Emilie Christiansen II	June 19, 2014	female	no	AB+	<a href="#">Profile</a>	<a href="#">Update</a>	
Fernando Pérez	June 26, 2018	undisclosed	no	*	<a href="#">Profile</a>	<a href="#">Update</a>	
Johnathon Fay V	July 29, 2016	undisclosed	no	AB+	<a href="#">Profile</a>	<a href="#">Update</a>	
Lenin Villacres	January 1, 1970	male	no	*	<a href="#">Profile</a>	<a href="#">Update</a>	
Maria Gleichner	May 28, 2010	female	no	A -	<a href="#">Profile</a>	<a href="#">Update</a>	

Fig. 19 Índice de Pacientes.  
Encalada, C. (junio 2018).

Los índices de usuarios y de pacientes son pantallas similares, sin embargo, contienen información pertinente al aspecto particular que gestionan, siendo éstos la información de cuenta y la información personal respectivamente.

## Gestión de Citas

Para gestionar citas, el usuario debe seleccionar la opción “Appointments” de la barra lateral (Fig. 9), a continuación, el sistema mostrará el Índice de Citas (Fig. 20) para listar las citas futuras agendadas en el sistema.

Appointments					Make Appointment	
Patient	Doctor	Date & Time	Type	Reason		
Johnathon Fay V	Mr. Nathaniel Jacobson	August 23, 2018 05:30	therapy	Rerum ratione illum veniam perferendis.	<a href="#">View</a>	<a href="#">Delete</a>
Prof. Garfield Hegmann	Maria Gleichner	August 16, 2018 18:54	therapy	Sed molestias at rerum qui aut cum.	<a href="#">View</a>	<a href="#">Delete</a>
Skyla Smith	Maria Gleichner	August 8, 2018 14:13	therapy	Voluptatem sed eos excepturi aut mollitia sunt.	<a href="#">View</a>	<a href="#">Delete</a>
Carlos Encalada	Maria Gleichner	July 30, 2018 03:36	check	Tenetur est commodi suscipit quidem aut.	<a href="#">View</a>	<a href="#">Delete</a>
Maria Gleichner	Carlos Encalada	July 19, 2018 05:02	consultation	Id beatae molestiae enim hic exercitationem qui odio suscipit.	<a href="#">View</a>	<a href="#">Delete</a>
Emilie Christiansen II	Maria Gleichner	July 13, 2018 13:13	check	Nam id consectetur dolore eos ut quos.	<a href="#">View</a>	<a href="#">Delete</a>

Fig. 20 Índice de Citas.  
Encalada, C. (junio 2018).

La creación de citas se puede realizar desde el Índice de Citas (Fig. 20) o desde los perfiles de paciente o doctor correspondientes según sea la necesidad del usuario. Cuando se crea una



cita desde un perfil determinado el sistema automáticamente completará los datos de usuario en el Formulario de Creación de Cita (Fig. 21), a continuación, el usuario deberá completar el resto de campos de manera que el sistema pueda llevar a cabo las validaciones pertinentes y finalmente registrar la cita para el doctor, paciente y hora seleccionados.

En la creación de citas, adicionalmente a la validación de integridad de datos, el sistema también validará automáticamente que la cita se encuentre a el horario del especialista seleccionado y que no traslape el tiempo asignado para una cita creada con anterioridad.

### Opciones del sistema

Todos los usuarios sin excepción tienen acceso al menú desplegable de opciones ubicado en la parte superior derecha de todas las pantallas (Fig. 21), en este menú se presentan las opciones del usuario para que pueda gestionar los datos de su cuenta personal, estas son:

- Mis Doctores, que corresponde a la lista de doctores con citas asignadas anteriormente para el paciente.
- Mis Citas, que corresponde a la lista de citas que el paciente tiene asignadas en el futuro.
- Resumen, si el usuario tuviese el rol de doctor, ésta opción lo llevaría a su perfil de médico.
- Perfil, que corresponde al perfil personal como paciente que tiene cada usuario.
- Cambiar contraseña, para actualizar la contraseña para inicio de sesión
- Cerrar sesión, para finalizar la sesión actual de modo que otro usuario pueda acceder al sistema desde el mismo equipo usando una cuenta diferente.

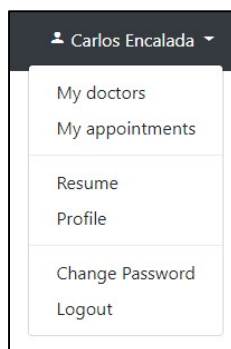


Fig. 21 Menú desplegable de Usuario.  
Encalada, C. (junio 2018).